PROGRAM SYSTEMS: THEORY AND APPLICATIONS ISSN 2079-3316
Research Article

UDC 004.932.75'1, 004.89
10.25209/2079-3316-2024-15-3-75-110

Recovering text sequences using deep learning models

Igor Victorovich **Vinokurov**
Financial University under the Government of the Russian Federation, Moscow, Russia
✉*igvvinokurov@fa.ru*

Abstract. This article presents the results of the formation, training and performance evaluation of models with the Encoder-Decoder and Sequence-To-Sequence (Seq2Seq) architectures for solving the problem of supplementing incomplete texts. Problems of this type often arise when restoring the contents of documents from their low-quality images. The studies conducted in the work are aimed at solving the practical problem of forming electronic copies of scanned documents of the «Roskadastr» PLC, the recognition of which is difficult or impossible with standard means.

The formation and study of models was carried out in Python using the high-level API of the Keras package. A dataset consisting of several thousand pairs was formed for the purpose of training and studying the models. Each pair in this set represented an incomplete and corresponding full text. To evaluate the quality of the models, the values of the loss function and the accuracy, BLEU and ROUGE-L metrics were calculated. Loss and accuracy made it possible to evaluate the effectiveness of the models at the level of predicting individual words. The BLEU and ROUGE-L metrics were used to evaluate the similarity between the full and reconstructed texts. The results showed that both the Encoder-Decoder and Seq2Seq models cope with the task of reconstructing text sequences from their fixed set, but the Seq2Seq transformer-based model achieves better results in terms of training speed and quality. (*Linked article texts in English and in Russian*).

Key words and phrases: deep learning models, encoder-decoder, sequence-to-sequence transformer, text recovering, BLEU, ROUGE-L, Keras, Python

2020 *Mathematics Subject Classification*: 68T20; 68T07, 68T45

For citation: Igor V. Vinokurov. *Recovering text sequences using deep learning models*. Program Systems: Theory and Applications, 2024, **15**:3(62), pp. 75–110. (*In English, in Russian*). https://psta.psiras.ru/read/psta2024_3_75-110.pdf

Introduction

In recent years, deep learning models (*Deep Neuaral Network*, DNN) have achieved significant results in the field of natural language processing (*Neural Language Processing*, NLP) [1]. Analysis of literature sources showed that the most common models used in such tasks as text transformation (translation), text recovery from distorted or incomprehensible documents, scanned documents of poor quality, illegible manuscripts, blurry or damaged images, etc. are Encoder-Decoder and Seq2Seq transformers.

The Encoder-Decoder architecture, based on recurrent neural networks (*Recurrent Neural Networks*, RNN) or convolutional neural networks (*Convolutional Neural Network*, CNN), consists of two main components – encoder and decoder [2]. The encoder transforms the input data into an internal representation taking into account key features of its content. The decoder uses this representation to generate output data by sequentially predicting its elements.

In contrast, the Seq2Seq transformer architecture [3] offers an alternative approach to representing sequences. It uses a transformation mechanism based on multiple layers with attention mechanisms [4], which allows this model to efficiently process long text sequences. The attention mechanism allows the model to take into account the importance of individual words in the context of the entire sentence, thereby contributing to the generation of higher-quality and coherent text. Compared with Encoder-Decoder, the Seq2Seq transformer has several advantages, such as better ability to handle long sequences, a more flexible architecture, and the ability to train models on large amounts of data.

To evaluate the quality of NLP models, one can use both conventional metrics loss, accuracy etc., and metrics specific to evaluating the quality of the generated text, the main ones being BLEU (*Bilingual Evaluation Understudy*) [5] and ROUGE-L (*Recall-Oriented Understudy for Gisting Evaluation – Longest Common Subsequence*) [6]. The first of them measures the similarity between the predicted and reference text. It uses syntactic information to compare sequences of n words (n-grams). The more matches in n-grams between the predicted and reference texts, the higher the BLEU

FIGURE 1. Document with highlighted sections of text. Text fragments recognized by OCR are highlighted in color

value. However, this metric does not take into account the semantic and contextual relationship between words, which may limit its applicability. The second ROUGE-L metric evaluates the quality of automatic text summarization. It compares the length of the longest common word sequence between the predicted and reference text with the length of the reference text, thereby measuring the coverage of the predicted text relative to the reference text and allows one to estimate the degree of information compression in the generated text.

The basis for conducting the research, the results of which are presented in this article, was the impossibility of restoring text on scanned documents of poor quality using modern OCR systems, Figure 1.

An obvious solution to this problem is to develop a simple sentence matching system. However, as noted above, DNN models are able to learn complex non-linear dependencies between input and output data, which allows them to more effectively model the context and semantics of text. In addition, DNN models can be more flexible and generalize, which makes them more effective when working with different types of text and information recovery tasks. *It is the generalizing properties of DNN models that served as the rationale for their use in solving the stated problem – a restored and semantically close sentence is better than its complete absence.*

Section 1 provides a rationale for the need for research and a statement of the problem. Section 2 is devoted to the analysis of works on restoring text sequences. Features of creating a dataset for training models and

researching their work are described in section 3. The creation and research of Encoder-Decoder and Seq2Seq models are given in section 4 and section 5 respectively. The advantages and disadvantages of restoring text sequences using the created models are given in section 6.

1. Settings the goal and research problems

When recognizing images of text documents using standard OCR tools, it is not always possible to obtain a high-quality copy in text format. The reason is blurry, illegible, or noisy (for example, in the form of handwritten notes) sections of text on the document image [7, 8].

The goal of this work is to research the applicability of the main DNN models for NLP – Encoder-Decoder and Seq2Seq transformer for restoring text from a given dataset.

The goal set in the work can be achieved by solving the following main problems:

(1) Creation of a dataset, consisting of preparing pairs in the form of incomplete and corresponding full text.

(2) Creation of optimal Encoder-Decoder and Seq2Seq transformer models for achieving the goal.

(3) Analysis of the quality of the models as a result of calculating the loss function and the accuracy, BLEU and ROUGE-L metrics.

2. Analysis of works on restoration of text sequences

An analysis of available publications has shown that there are two main types of models that can be used to match and restore text sequences.

(1) Attention-based models (in most cases, these are Seq2Seq transformer models) that can focus on the context of a sentence and select the most appropriate option for replacing or restoring missing words.

(2) RNN- and CNN-based models that take into account the context of a sentence and, in the case of CNN, the features of its visualization on images.

Below is a description of the most significant, in the author's opinion, works on the use of these types of models for restoring the original text.

In [9], the authors consider and demonstrate the effectiveness of two classes of attention mechanisms for matching texts in different languages. The first takes into account all source words, the second considers only a subset of the source words.

Missing word recovery using models based on variational autoencoders (VAE) is proposed, for example, in [10]. Here, VAEs implement prediction of the input text sequence for subsequent improvement of RNN training.

The Seq2Seq model, which allows to perform the task of generating corrected text using the attention mechanism, is described in [11]. The model can be used to transform an incorrect, incorrect sequence of characters into corrected text.

A model based on the Seq2Seq architecture with an attention mechanism for error correction in text obtained from an OCR system is given in [13]. The model is trained on a sufficiently large number of text pairs recognized using OCR.

A study of the applicability of BERT (Bidirectional Encoder Representations from Transformers) models for error correction in sentences of non-English native speakers is given in [14]. The model is able to use the context of a sentence to correct grammatical and stylistic errors in the text.

A pre-trained model with the architecture of a masked Seq2Seq transformer for reconstructing a text fragment from its remaining part is considered in [15]. The model's encoder receives a sentence with a randomly masked fragment (several consecutive tokens) as input, and its decoder tries to predict this masked fragment. The paper shows that as a result of fine-tuning, the model is able to reconstruct the original text quite accurately.

The authors of the paper [16] propose a method for correcting errors in text using a neural network based on symbolic self-attention. The model uses the character level to more accurately correct typos, spelling, and other types of errors in the text.

In [17], the author describes the use of RNN for language models and proposes methods for generating text in a given context. The publication is devoted to controlled sequence labelling – an important area of machine learning, including such tasks as speech recognition, handwriting recognition, and part-of-speech labelling.

The study of the applicability of neural network models for restoring distorted character images is carried out in [18]. The authors describe methods for restoring damaged text based on a combination of CNN and RNN.

The article [19] presents a CNN-based model for automatically correcting typos in text. The model is trained on a large corpus of texts and is able to correct typos with high accuracy.

In [20], a method for restoring distorted images using CNN without the need for training on a large dataset is presented. The method was originally developed for image restoration, but can also be applied to restoring damaged text.

The authors of the article [21] present an autoencoder architecture for error correction in OCR systems. The autoencoder model is used to extract hidden text representations and then restore them.

In addition to the above, a fairly large number of works on restoring text sequences are devoted to restoring texts on different types of images of historical documents damaged over time. Their detailed analysis is given in [22].

3. Dataset creation

To train the Encoder-Decoder and Seq2Seq models and to research their operation, a proprietary dataset was created, which, as noted above, is a set of pairs of the form:

incomplete text $\rightarrow$ [*start*]full text[*end*]

The main types of documents of the «Roscadastre» PLC software and software package were used to form the dataset. All unique sentences of these documents in the created dataset represent the full text. The

Полный текст (электронный документ)
Сведения об уточняемых земельных участках и их частях

Неполный текст (результат распознавания OCR)
1. Сведения об
2. об уточняемых
3. Сведения об уточняемых
4. Сведения об уточняемых земельных
5. об уточняемых земельных
6. Сведения об уточняемых земельных участках
7. уточняемых земельных участках
8. Сведения об уточняемых земельных участках и их
9. земельных участках и их частях
10. об уточняемых земельных участках и
...

FIGURE 2. Full text and its corresponding first 10 incomplete texts

results of removing continuous sequences of 1 to $N-3$ words from the full text form a set of corresponding incomplete texts (here N is the number of words in the text). The removal of continuous sequential words from the text is explained by the specifics of blurring and the location of illegible sections of text on scanned documents, see Figure 1.

The number of full-text sentences included in the created dataset does not exceed 250, the number of corresponding incomplete sentences included in the dataset is about 3000. An example of the correspondence between them is shown in Figure 2.

Tokenization and vectorization of text pairs was carried out using TextVectorization from the Keras package. To remove ambiguity when restoring full texts, text standardization before tokenization involved the inclusion of a feature of the document area in which it may be present.

For training the DNN, 80% of the pairs from the created dataset were used, for validation and testing – 10%.

4. Creation and research of the Encoder-Decoder model

The Encoder-Decoder model was developed and studied in Python using the API Keras [**23**–**25**]. Figure 3 shows the optimal structure of the model obtained as a result of experimental studies.

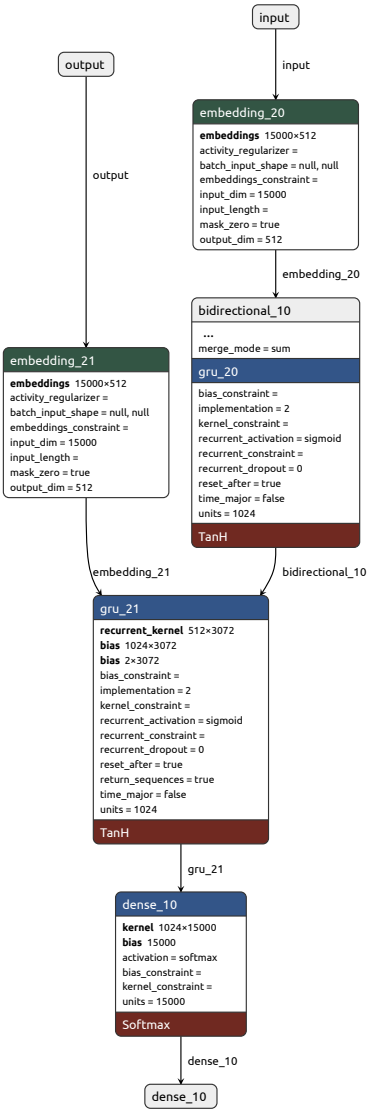

FIGURE 3. Encoder-Decoder model in Keras API for partial text recovery

The encoder of this model consists of the following layers:

(1) Input layer (InputLayer), takes input data in the form of text sequences.
(2) Embedding layer (Embedding) transforms words of text sequences into their vector representations.
(3) Recurrent layer (BidirectionalGRU) processes sequences in both forward and backward directions and produces a hidden state (context vector).

Decoder model layers:

(1) Input layer (InputLayer) takes input as a sequence of words. Determines the shape and type of the input data.
(2) The embedding layer (Embedding) transforms the input tokens into dense vector representations of a given dimensionality, similar to the encoder.
(3) The recurrent layer (GRU) takes the vector representations and processes them sequentially, generating the full text, taking into account the context from the encoder. It is initialized with the state generated by the encoder.
(4) The output layer (Dense) takes the output from the decoder at the current time step and transforms it into a vector of probabilities, each component of which corresponds to a possible next token of the reconstructed text.

The description of the main parameters of all layers of this model is given in table 1.

To assess the quality of the model at the level of individual words of text sequences, the values of the loss function and the accuracy metric were calculated. Figure 4 and Figure 5 show the values of the loss function sparse_categorical_crossentropy and the metric accuracy of the model for 30 training epochs. The number of epochs was found experimentally and is optimal. The stochastic optimization algorithm rmsprop was used to train the model.

The average values of the BLEU and ROUGE-L metrics, which allow us to evaluate the accuracy of reconstructing incomplete text from the testing dataset, are shown in Figure 6. Values in the range of 0.3-0.4 correspond to understanding and acceptable translation of the text. To calculate the metric values, the functions sentence_bleu() and get_scores() from the NLTK and Rouge packages were used, respectively. They were called after the completion of each training epoch from the callback functions of the fit() method.

TABLE 1. Layers of the Encoder-Decoder Model

Layer type (title in Figure 3)	Activity function	Input tensor	Output tensor
InputLayer (input)	–	[(None, None)]	[(None, None)]
Embedding (embedding_20)	–	(None, None)	(None, None, 512)
Bidirectional(GRU) (bidirectional_10)	tanh	(None, None, 512)	(None, 1024)
InputLayer (output)	–	[(None, None)]	[(None, None)]
Embedding (embedding_21)	–	(None, None)	(None, None, 512)
GRU (gru_21)	tanh	[(None, None, 512), (None, 1024)])	(None, None, 1024)
Dense (dense_10)	softmax	(None, None, 1024)	(None, None, 15000)

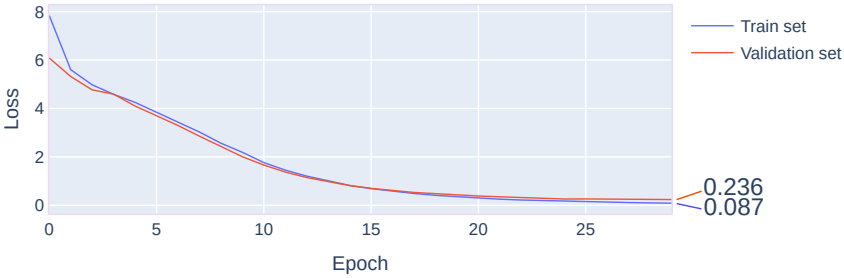

FIGURE 4. Model Losses

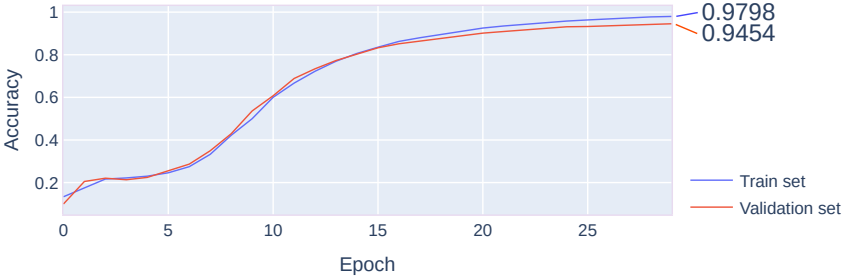

FIGURE 5. Model accuracy

Figure 7 shows the results of text recovery from 10 randomly generated datasets for testing. Each test set consisted of 250–300 sentences. The

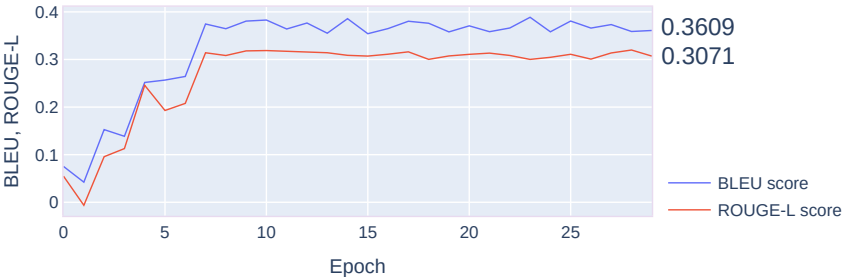

FIGURE 6. Metrics for evaluating the quality of text recovery

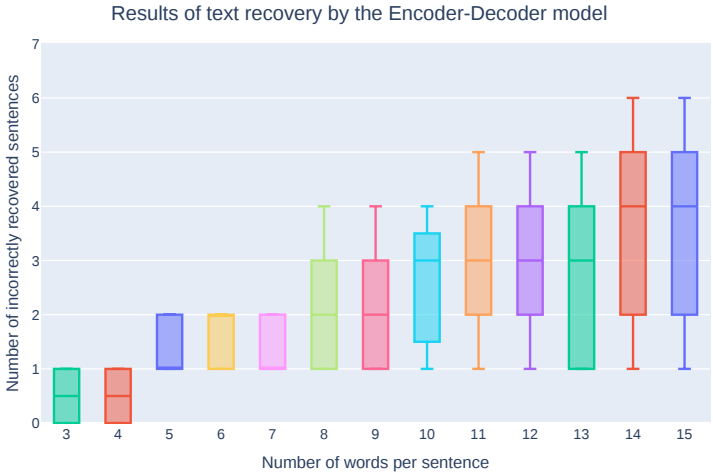

FIGURE 7. Numbers of incorrectly recovered sentences by the Encoder-Decoder model from 10 test datasets

number of incorrectly reconstructed sentences consisting of 11-15 words ranged from 1 to 6.

5. Creation and research of the Seq2Seq transformer model

As with the previous model, the creation and research of the model with the Seq2Seq architecture was carried out in Python using the API Keras [23–25].

The optimal structure of the Seq2Seq model, obtained as a result of experimental studies, is shown in Figure 8.

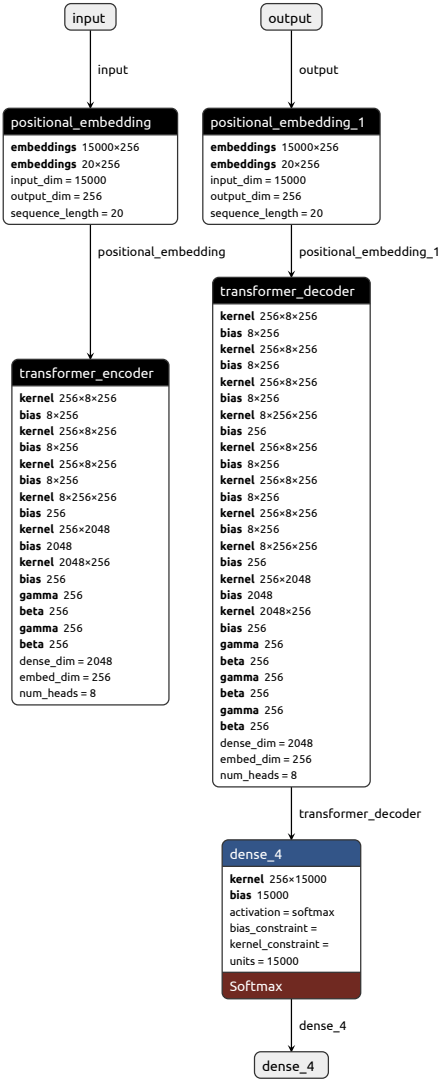

FIGURE 8. Encoder and decoder of the Seq2Seq transformer model in the Keras API for partial text recovery

TABLE 2. Layers of the Seq2Seq transformer model

Layer type (title in Figure 8)	Activity function	Input tensor	Output tensor
InputLayer (input)	–	[(None, None)]	[(None, None)]
PositionalEmbedding (positional_embedding)	–	(None, None)	(None, None, 256)
TransformerEncoder (transformer_encoder)	relu	(None, None, 256)	(None, None, 256)
InputLayer (output)	–	[(None, None)]	[(None, None)]
PositionalEmbedding (positional_embedding_1)	–	(None, None)	(None, None, 256)
TransformerDecoder (transformer_decoder)	relu	(None, None, 256)	(None, None, 256)
Dense (dense_4)	softmax	(None, None, 256)	(None, None, 15000)

The transformer model consists of the following layers:

(1) The Input Layer (InputLayer) takes input as a sequence of words. It determines the shape and type of the input data.

(2) Positional Embedding Layer (Positional Embedding Layer) adds information about the position of a word in a sequence. This allows the model to take into account the order of words in the input and output sequences.

(3) Transformer Encoder Layers (TransformerEncoder Layer), each of which implements an attention mechanism and contains fully connected layers. Due to this, they allow modeling dependencies in sequences, extracting features from the input data and representing them in an optimal internal representation for more complex computations and natural language processing tasks.

(4) Transformer Decoder Layers (TransformerDecoder Layer). Similar to the encoder, the decoder consists of several transformer decoder layers, which also include an attention mechanism and fully connected layers. The decoder generates an output sequence based on the context representations of the encoder.

(5) The output layer (Dense) transforms the predicted token representations into a probability distribution over all possible tokens.

To create more efficient and related representations of text sequences, the attention mechanism [4] is implemented in the encoder and decoder layers of the Seq2Seq architecture model. Table 2 provides a description of the main parameters of all layers of this model.

The study of the accuracy of the model, by analogy with the previous

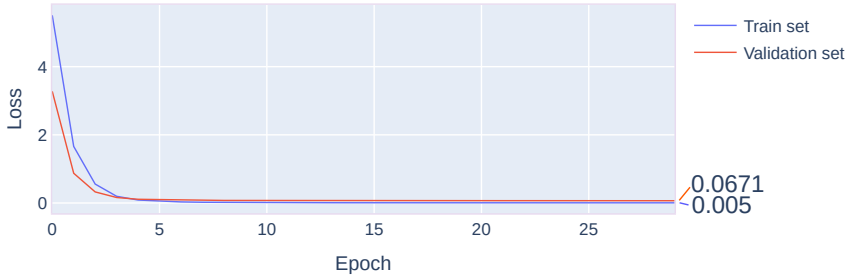

FIGURE 9. Model Losses

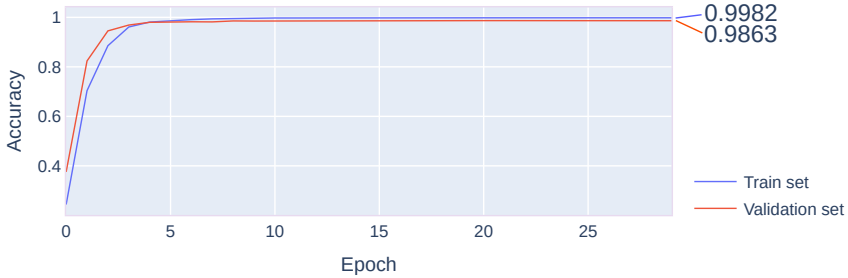

FIGURE 10. Model accuracy

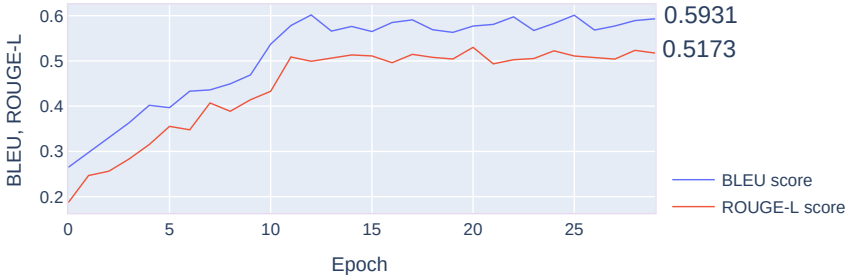

FIGURE 11. Metrics for evaluating the quality of text recovery

one, consisted in calculating the values of the loss function and the accuracy metric – sparse_categorical_crossentropy and accuracy respectively for each of the 30 epochs of its training, Figure 9, 10. The number of epochs was found experimentally and is optimal. When training the model, the stochastic optimization algorithm rmsprop was used.

The average values of the BLEU and ROUGE-L metrics, which allow us to evaluate the accuracy of the model in reconstructing the incomplete text from the testing dataset, are shown in Figure 11. Values from the range of 0.5-0.6 correspond to high quality of text translation.

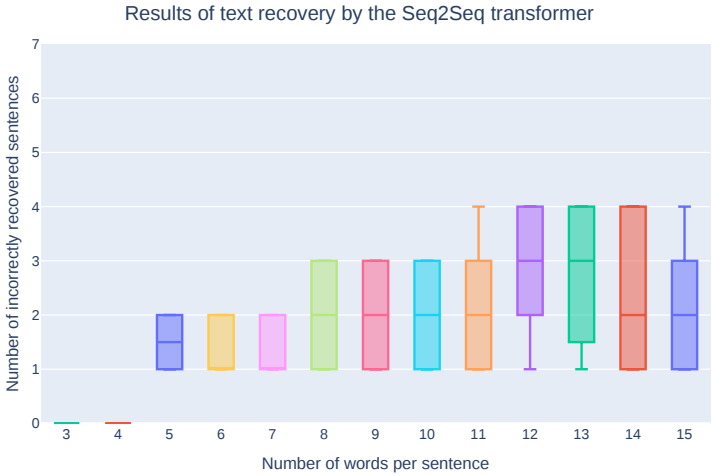

FIGURE 12. Numbers of incorrectly recovered sentences by Seq2Seq model from 10 test datasets

Figure 12 shows the results of incomplete text recovery from 10 randomly generated testing datasets. The number of incorrectly recovered sentences is less than for the previous model and ranges from 1 to 4. As a result, it is worth noting the quite natural result – the Seq2Seq transformer model is more efficient than the Encoder-Decoder model due to the use of the attention mechanism. The attention mechanism allows the transformer to highlight key elements of input and output sequences and more effectively identify long-term dependencies in them, which makes the model capable of learning on complex text generation (in this case, text recovery) tasks.

6. Conclusions, advantages and disadvantages of the proposed approach

The studies conducted in the work showed that the use of DNN models allows solving the problem posed in the work quite effectively. The results of text recovery from documents of the «Roscadastre» PLC acceptable for solving the practical problem are explained by the features of the dataset formation – it contains all pairs with incomplete and corresponding full texts. In addition, the incomplete text was considered in this work as a sequence of a certain number of adjacent words, which significantly simplified the process of its comparison with the full text. Sequences of arbitrary words of the full text were not included in the dataset.

The advantage of the proposed approach is the simplicity of the models and the features of the dataset formation for their training and validation.

FIGURE 13. Correctly and incorrectly restored text "Information on clarified" (see Figure 1 and Figure 2)

TABLE 3. Metric values of Encoder-Decoder and Seq2Seq transformer models

Loss metric	Accuracy metric	BLEU metric	ROUGE-L metric
Encoder-Decoder model			
0.087	0.9798	0.3609	0.3071
Seq2Seq transformer model			
0.005	0.9982	0.5931	0.5173

The disadvantage is the impossibility of restoring the text from a set of its arbitrary (inconsistent) words without significantly complicating the model, which involves analyzing the context of the sentence. The only reason for incorrect text recovery is related to the rather rare cases (1-3% of the total number) of the formation of identical embeddings (vectorization of incomplete text and the indicator of the document area in which the corresponding full text may be found, see section 3). An example of correct and incorrect recovery of incomplete text with identical embeddings is shown in Figure 13.

The results obtained in this work are planned for use in the information system of the «Roscadstr» PLC control and processing center for the purpose of converting scanned documents into their text analogues. To implement this process, the Seq2Seq transformer model was selected as it showed the best result compared to the Encoder-Decoder model, table. 3.

The metrics provided in this table are relevant only to the dataset created from the documents of the «Roscadastr» PLC. They may differ for datasets of other subject areas.

References

[1] N. C. Sabharwal, A. Agrawal. *Hands-on Question Answering Systems with BERT: Applications in Neural Networks and Natural Language Processing*, Apress, Berkeley, CA, 2021, ISBN 978-1-4842-6664-9, xv+184 pp. ↑76

[2] K. Aitken, V V. Ramasesh, Y. Cao, N. Maheswaranathan. *Understanding how encoder-decoder architectures attend*, 2021, 24 pp. arXiv 2110.15253~[cs.LG] ↑76

[3] A. Rahali, M. A. Akhloufi. "End-to-end transformer-based models in textual-based NLP", *Artificial Intelligence*, **4**:1 (2023), pp. 54–110. ↑76

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. *Attention is all you need*, 2017, 15 pp. arXiv 1706.03762 ↑76, 87

[5] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu. “BLEU: a method for automatic evaluation of machine translation”, *ACL’02 Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (July 7–12, 2002, Philadelphia, Pennsylvania, USA), ACL, Stroudsburg, 2002, pp. 311–318. ↑76

[6] Ch.-Y. Lin. “ROUGE: a package for automatic evaluation of summaries”, *Proceedings of the Workshop on Text Summarization Branches Out*, WAS 2004 (July, 2004, Barcelona, Spain), ACL, 2004, 74–81 pp. ↑76

[7] Vinokurov I. V.. “Using a convolutional neural network to recognize text elements in poor quality scanned images”, *Program Systems: Theory and Applications*, **13**:3(54) (2022), pp. 45–59. ↑78

[8] Vinokurov I. V.. “Recognition of digital sequences using convolutional neural networks”, *Program Systems: Theory and Applications*, **14**:3 (2023), pp. 3–36 (in Russian, in English). ↑78

[9] Th. Luong, H. Pham, Ch. D. Manning. “Effective approaches to attention-based neural machine translation”, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (17–21 September, 2015, Lisbon, Portugal), ACL, 2015, ISBN 978-1-941643-32-7, pp. 1412–1421. ↑79

[10] A. M. Dai, Q. V. Le. *Semi-supervised Sequence Learning*, NIPS 2015 (December 7–12, 2015, Montreal, Quebec, Canada), Advances in Neural Information Processing Systems, vol. **28**, Curran Associates, Inc., 2015, ISBN 9781510825024, 9 pp. ↑79

[11] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. N. Dauphin. “Convolutional sequence to sequence learning”, *Proceedings of the 34th International Conference on Machine Learning* (6–11 August 2017, International Convention Centre, Sydney, Australia), PMLR, vol. **70**, 2017, pp. 1243–1252. ↑79

[12] D. Ulyanov, A. Vedaldi, V. Lempitsky. “Deep image prior”, *International Journal of Computer Vision*, **128**:7 (2020), pp. 1867–1888. ↑

[13] K. Hakala, A. Vesanto, N. Miekka, T. Salakoski, F. Ginter. *Leveraging text repetitions and denoising autoencoders in OCR post-correction*, 2019, 5 pp. arXiv 1906.10907~[cs.CL] ↑79

[14] G. Huang, J. Wang, H. Tang, X. Ye. “BERT-based contextual semantic analysis for English preposition error correction”, *Journal of Physics: Conference Series*, **1693**:1 (2020), id. 012115, 5 pp. ↑79

[15] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu. “MASS: masked sequence to sequence pre-training for language generation”, International Conference on Machine Learning (9–15 June 2019, Long Beach, California, USA), PMLR, vol. **97**, 2019, pp. 5926–5936. arXiv 1905.02450~[cs.CL] ↑79

[16] Sh. Chollampatt, D, T. Hoang, H. T. Ng. “Adapting grammatical error correction based on the native language of writers with neural network joint models”, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2016 (1–4 November, 2016, Austin, Texas, USA), ACL, 2016, ISBN 978-1-945626-25-8, pp. 1901–1911. ↑79

[17] A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*, Studies in Computational Intelligence, vol. **385**, Springer, Berlin–Heidelberg, 2012, ISBN 978-3-642-24797-2, 146 pp. ↑80

[18] T. Ge, X. Zhang, F. Wei, M. Zhou. "Automatic grammatical error correction for sequence-to-sequence text generation: an empirical study", *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (July 28–August 2, 2019, Florence, Italy), ACL, 2019, ISBN 978-1-950737-48-2, pp. 6059–6064. ↑80

[19] X. Zhang, J. Zhao, Y. LeCun. "Character-level convolutional networks for text classification", 2016, 9 pp. arXiv 1509.01626 [cs.LG] ↑80

[20] Z. Xie, A. Avati, N. Arivazhagan, D. Jurafsky, A. Ng. *Neural language correction with character-based attention*, 2016, 10 pp. arXiv 1603.09727 [cs.CL] ↑80

[21] J. Ramirez-Orta, E. Xamena, A. Maguitman, E. Milios, A. Soto. "Post-OCR document correction with large ensembles of character sequence-to-sequence models", *Proceedings of the AAAI Conference on Artificial Intelligence*, **36** (2022), pp. 11192–11199. ↑80

[22] A. A. Alkhazraji, K. Baheeja, A. M. N. Alzubaidi. "Ancient textual restoration using deep neural networks: a literature review", *2023 Al-Sadiq International Conference on Communication and Information Technology*, AICCIT 2023 (04–06 July 2023, Al-Muthana, Iraq), 2023, ISBN 9798350341898, pp. 64–69. ↑80

[23] F. Chollet. *Deep Learning with Python*, 2nd ed., Manning, 2021, ISBN 9781617296864, 504 pp. ↑81, 85

[24] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed., O'Reilly Media, Sebastopol, 2019, ISBN 978-1-492-03264-9, 848 pp. ↑81, 85

[25] A. Kapoor, A. Gulli, S. Pal. *Deep Learning with TensorFlow and Keras: Build and deploy supervised, unsupervised, deep, and reinforcement learning models*, 3rd ed., Packt Publishing, 2022, ISBN 978-1803232911, 698 pp. ↑81, 85

Received 03.03.2024;
approved after reviewing 14.04.2024;
accepted for publication 15.08.2024;
published online 23.09.2024.

Information about the author:

Igor Victorovich Vinokurov

Candidate of Technical Sciences (PhD), Associate Professor at the Financial University under the Government of the Russian Federation. Research interests: information systems, information technologies, data processing technologies

0000-0001-8697-1032
e-mail: igvvinokurov@fa.ru

The author declare no conflicts of interests.

RU EN ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ ISSN 2079-3316
научная статья

УДК 004.932.75'1, 004.89
10.25209/2079-3316-2024-15-3-75-110

Восстановление текстовых последовательностей с использованием моделей глубокого обучения

Игорь Викторович **Винокуров**✉
Финансовый Университет при Правительстве Российской Федерации, Москва, Россия
✉*igvvinokurov@fa.ru*

Аннотация. В статье приведены результаты формирования, обучения и оценки качества работы моделей с архитектурами Encoder-Decoder и Sequence-To-Sequence (Seq2Seq) для решения задачи дополнения неполных текстов. Задачи такого типа достаточно часто возникают при восстановлении содержимого документов по их некачественным изображениям. Проведённые в работе исследования ориентированы на решение практической задачи формирования электронных копий отсканированных документов ППК «Роскадастр», распознавание которых стандартными средствами затруднительно или невозможно.

Формирование и исследование моделей осуществлялось на языке Python с использованием высокоуровневого API пакета Keras. С целью обучения и исследования моделей был сформирован набор данных, состоящий из нескольких тысяч пар. Каждая пара этого набора представляла собой неполный и соответствующий ему полный тексты. Для оценки качества работы моделей осуществлялось вычисление значений функции потерь loss и метрик accuracy, BLEU и ROUGE-L. Loss и accuracy позволили оценить эффективность моделей на уровне предсказания отдельных слов. Метрики BLEU и ROUGE-L использовались для оценки сходства между полными и восстановленными текстами. Полученные результаты показали, что обе модели Encoder-Decoder и Seq2Seq справляются с задачей восстановления текстовых последовательностей из их фиксированного множества, однако модель на основе трансформера Seq2Seq позволяет достичь лучших результатов по скорости и качеству обучения. (*Связанные тексты статьи на английском и на русском языках*)

Ключевые слова и фразы: модели глубокого обучения, Encoder-Decoder, трансформер Sequence-To-Sequence, восстановление текста, BLEU, ROUGE-L, Keras, Python

Для цитирования: Винокуров И. В. *Восстановление текстовых последовательностей с использованием моделей глубокого обучения* // Программные системы: теория и приложения. 2024. **Т. 15**. № 3(62). С. 75–110. (*Англ.+русс.*)
https://psta.psiras.ru/read/psta2024_3_75-110.pdf

Введение

В последние годы с помощью моделей глубокого обучения (*Deep Neuaral Network*, DNN) получены существенные результаты в области обработке естественного текста (*Neural Language Processing*, NLP) [1]. Анализ литературных источников показал, что самыми распространёнными моделями, используемыми в таких задачах как преобразование (перевод) текста, восстановления текста из искаженных или непонятных документов, отсканированных документов плохого качества, нечитаемых рукописей, размытых или поврежденных изображений и т.п. являются Encoder-Decoder и трансформеры Seq2Seq.

Архитектура Encoder-Decoder, основанная на рекуррентных нейронных сетях (*Recurrent Neural Networks*, RNN) или свёрточных нейронных сетях (*Convolutional Neural Network*, CNN), состоит из двух основных компонентов – энкодера и декодера [2]. Энкодер преобразует входные данные во внутреннее представление, учитывая ключевые особенности их содержания. Декодер использует это представление для генерации выходных данных, последовательно предсказывая их элементы.

В отличие от этой модели, архитектура трансформера Seq2Seq [3] предлагает альтернативный подход к представлению последовательностей. Она использует механизм трансформации, основанный на множестве слоёв с механизмами внимания [4], что позволяет этой модели эффективно обрабатывать длинные текстовые последовательности. Механизм внимания позволяет модели учитывать важность отдельных слов в контексте всего предложения, способствуя тем самым генерации более качественного и связного текста. По сравнению с Encoder-Decoder, трансформер Seq2Seq обладает рядом преимуществ, таких как лучшая способность обращаться с длинными последовательностями, более гибкая архитектура и возможность обучения моделей на больших объёмах данных.

Для оценки качества моделей в NLP могут быть использованы как обычные метрики loss, accuracy и т.п., так и метрики, специфичные для оценки качества сгенерированного текста, основными из которых являются BLEU (*Bilingual Evaluation Understudy*) [5] и ROUGE-L (*Recall-Oriented Understudy for Gisting Evaluation – Longest Common Subsequence*) [6]. Первая из них измеряет схожесть между предсказанным и эталонным текстом. Она использует синтаксическую информацию для сравнения последовательностей из n слов (n-грамм). Чем больше совпадений в n-граммах между предсказанным и эталонным текстами, тем выше будет значение

BLEU. Однако, эта метрика не учитывает семантическую и контекстную связь между словами, что может ограничивать её применимость. Вторая из метрик ROUGE-L оценивает качество автоматической суммаризации текста. Она сравнивает длину наибольшей общей последовательности слов между предсказанным и эталонным текстом с длиной эталонного текста, измеряя тем самым покрытие предсказанного текста относительно эталонного и позволяет оценить степень сжатия информации в сгенерированном тексте.

Основанием для проведения исследований, результаты которых приведены в этой статье, явилось невозможность восстановления текста на отсканированных документах плохого качества современными OCR-системами, рисунок 1.

Рисунок 1. Документ с осветлёнными участками текста. Фрагменты текста, распознаваемые OCR, выделены цветом

Очевидным решением этой задачи является разработка простой системы соответствия предложений. Однако, как было отмечено выше, DNN-модели способны изучать сложные нелинейные зависимости между входными и выходными данными, что позволяет им более эффективно моделировать контекст и семантику текста. Кроме того, DNN-модели могут быть более гибкими и способными к обобщению, что делает их более эффективными при работе с различными типами текста и задачами восстановления информации. *Именно наличие у DNN-моделей обобщающих свойств послужило обоснованием для их использования при решении поставленной задачи – восстановленное и близкое по смыслу предложение лучше, чем его полное отсутствие.*

В разделе 1 осуществляется обоснование необходимости исследований и постановка задачи. Раздел 2 посвящён анализу работ по восстановлению

текстовых последовательностей. Создание набора данных для обучения моделей описано в разделе 3. Формирование и исследование моделей Encoder-Decoder и Seq2Seq приведено в разделе 4 и разделе 5 соответственно. Достоинства и недостатки восстановления текстовых последовательностей с использованием сформированных моделей приведены в разделе 6.

1. Постановка цели и задач исследования

При распознавании изображений текстовых документов стандартными средствами OCR не всегда можно получить их качественную копию в текстовом формате. Причиной являются размытые, неразборчивые или зашумлённые (например, в виде заметок от руки) участки текста на изображении документа [7, 8].

Целью данной работы является исследование применимости основных DNN-моделей для NLP – Encoder-Decoder и трансформера Seq2Seq для восстановления текста из заданного набора данных.

Поставленная в работе цель может быть достигнута за счёт решения следующих основных задач:

(*1*) Формирование набора данных, заключающееся в подготовке пар в виде неполного и соответствующего ему полного текста.

(*2*) Формирование оптимальных для достижения поставленной цели моделей Encoder-Decoder и трансформера Seq2Seq.

(*3*) Анализ качества работы моделей в результате вычисления функции потерь и метрик accuracy, BLEU и ROUGE-L.

2. Анализ работ по восстановлению текстовых последовательностей

Анализ доступных публикаций показал, что существует два основных типа моделей, которые могут быть использованы для сопоставления и восстановления текстовых последовательностей.

(*1*) Модели с механизмом внимания (в большинстве случаев это модели трансформеров Seq2Seq), способные сфокусироваться на контексте предложения и выбрать наиболее подходящий вариант для замены или восстановления пропущенных слов.

(*2*) Модели на основе RNN и CNN, позволяющие учитывать контекст предложения и, в случае CNN, особенности его визуализации на изображения.

Ниже приведено описание наиболее значимых, по мнению автора, работ по использованию моделей этих типов для восстановления оригинального текста.

В [9] авторы рассматривают и демонстрируют эффективность двух классов механизма внимания для сопоставления текстов на разных языках. Первый учитывает все исходные слова, второй рассматривает только подмножество исходных слов.

Восстановления пропущенных слов с использованием моделей на основе вариационных автоэнкодеров (*Variational Autoencoder*, VAE) предлагается, например, в [10]. Здесь VAE реализуют прогнозирование входной текстовой последовательности для последующего улучшения обучения RNN.

Модель Seq2Seq, позволяющая выполнить задачу генерации исправленного текста с использованием механизма внимания, описана в [11]. Модель может использоваться для преобразования неверной, некорректной последовательности символов в исправленный текст.

Модель на основе архитектуры Seq2Seq с механизмом внимания для исправления ошибок в тексте, полученном из OCR системы, приведена в [13]. Модель обучается на достаточно большом количестве текстовых пар, распознанных с помощью OCR.

Исследование применимости моделей BERT (*Bidirectional Encoder Representations from Transformers*) для исправления ошибок в предложениях неанглийских носителей языка, приведено в [14]. Модель способна использовать контекст предложения для исправления грамматических и стилистических ошибок в тексте.

Предварительно обученная модель с архитектурой маскированного трансформера Seq2Seq для восстановления фрагмента текста по его оставшейся части рассматривается в [15]. Кодер модели принимает на вход предложение со случайно замаскированным фрагментом (несколько последовательных токенов), а его декодер пытается предсказать этот замаскированный фрагмент. В работе показано, что в результате тонкой настройки модель способна достаточно точно восстанавливать исходных текст.

Авторы статьи [16] предлагают метод исправления ошибок в тексте с помощью нейронной сети, основанной на символьном самовнимании. Модель использует уровень символов для более точного исправления опечаток, орфографических и других типов ошибок в тексте.

В [17] автор описывает использование RNN для языковых моделей и предлагает методы для генерации текста в заданном контексте. Издание посвящено маркировке контролируемых последовательностей – важной области машинного обучения, включающей такие задачи как распознавание речи, рукописного ввода и маркировка частей речи.

Исследование применимости нейросетевых моделей для восстановления искажённых символьных изображений проводится в [18]. Авторы описывают методы восстановления испорченного текста, основанные на сочетании CNN и RNN.

В статье [19] представлена модель на основе CNN для автоматического исправления опечаток в тексте. Модель обучается на большом корпусе текстов и способна исправлять опечатки с высокой точностью.

В [20] приведён метод восстановления искажённых изображений с использованием CNN без необходимости обучения на большом наборе данных. Метод был изначально разработан для восстановления изображений, но может быть также применен и к восстановлению испорченного текста.

Авторы статьи [21] представляют архитектуру автоэнкодера для исправления ошибок в OCR-системах. Модель с автоэнкодером используется для извлечения скрытых представлений текста и их последующего восстановления.

Помимо указанных выше, достаточно большое количество работ по восстановлению текстовых последовательностей посвящено восстановлению текстов на разного типа изображениях исторических документов, повреждённых с течением времени. Их подробный анализ приведён в [22].

3. Формирование набора данных

Для обучения моделей Encoder-Decoder и Seq2Seq и последующего исследования их работы был сформирован собственный набор данных, представляющий собой, как уже отмечалось выше, множество пар вида:

$$\text{неполный текст} \rightarrow [start]\text{полный текст}[end]$$

Для формирования датасета использовались основные типы документов ППК «Роскадастр». Все уникальные предложения этих документов в сформированном датасете представляют собой полный текст. Результаты удаления из полного текста непрерывных последовательностей от 1

до N-3 слов образует совокупность соответствующих ему неполных текстов (здесь N – количество слов в тексте). Удаление из текста именно непрерывных последовательных слов объясняется спецификой размытия и расположением неразборчивых участков текста на отсканированных документах, см. рисунок 1.

Количество предложений с полным текстом, входящих в сформированный датасет не превышает 250, количество соответствующих им неполных предложений, вошедших в датасет, составляет порядка 3000. Пример соответствия одного другому приведён на рисунке 2.

Полный текст (электронный документ)
Сведения об уточняемых земельных участках и их частях

Неполный текст (результат распознавания OCR)
1. Сведения об
2. об уточняемых
3. Сведения об уточняемых
4. Сведения об уточняемых земельных
5. об уточняемых земельных
6. Сведения об уточняемых земельных участках
7. уточняемых земельных участках
8. Сведения об уточняемых земельных участках и их
9. земельных участках и их частях
10. об уточняемых земельных участках и

...

Рисунок 2. Полный текст и соответствующие ему первые 10 неполных текстов

Токенизация и векторизации текстовых пар осуществлялась с использованием TextVectorization из пакета Keras. Для снятия неоднозначности при восстановлении полных текстов, стандартизация текста перед токенизацией предполагала включение признака области документа, в которой он может присутствовать.

Для обучения DNN использовалось 80% пар из сформированного датасета, для валидации и тестирования – по 10%.

4. Формирование и исследование модели Encoder-Decoder

Формирование и исследование модели Encoder-Decoder осуществлялось на языке Python с использованием API Keras [**23**–**25**]. На рисунке 3 приведена оптимальная структура модели, полученная в результате проведения экспериментальных исследований.

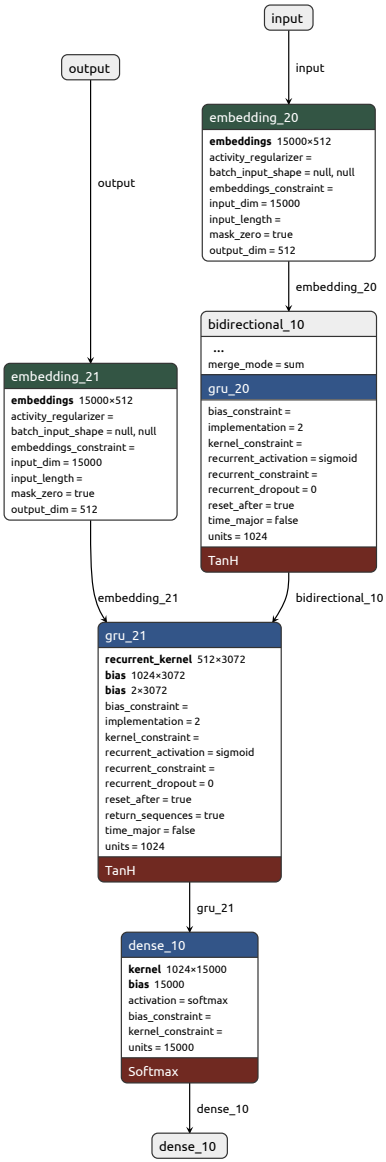

Рисунок 3. Модель Encoder-Decoder в API Keras для восстановления неполного текста

Энкодер этой модели состоит из следующих слоёв:

(1) Входной слой (InputLayer), принимает входные данные в виде текстовых последовательностей.

(2) Слой эмбеддинга (Embedding), преобразует слова текстовых последовательностей в их векторные представления.

(3) Рекуррентный слой (BidirectionalGRU) обрабатывает последовательности и в прямом и обратном направлениях и выдаёт скрытое состояние (вектор контекста).

Слои декодера модели:

(1) Входной слой (InputLayer) принимает входные данные в виде последовательности слов. Определяет форму и тип входных данных.

(2) Слой эмбеддинга (Embedding) преобразует входные токены в плотные векторные представления заданной размерности, аналогично энкодеру.

(3) Рекуррентный слой (GRU) принимает векторные представления и последовательно обрабатывает их, генерируя полный текст и учитывая контекст из энкодера. Инициализируется состоянием, сгенерированным энкодером.

(4) Выходной слой (Dense) принимает выходные данные из декодера в текущем временном шаге и преобразует их в вектор вероятностей, каждая компонента которого относится к возможному следующему токену восстановленного текста.

Описание основных параметров всех слоёв этой модели приведено в таблице 1.

Таблица 1. Слои модели Encoder-Decoder

Тип слоя (имя на рисунке 3)	Функция активации	Входной тензор	Выходной тензор
InputLayer (input)	–	[(None, None)]	[(None, None)]
Embedding (embedding_20)	–	(None, None)	(None, None, 512)
Bidirectional(GRU) (bidirectional_10)	tanh	(None, None, 512)	(None, 1024)
InputLayer (output)	–	[(None, None)]	[(None, None)]
Embedding (embedding_21)	–	(None, None)	(None, None, 512)
GRU (gru_21)	tanh	[(None, None, 512), (None, 1024)])	(None, None, 1024)
Dense (dense_10)	softmax	(None, None, 1024)	(None, None, 15000)

Для оценки качества работы модели на уровне отдельных слов текстовых последовательностей осуществлялось вычисление значений функции потерь и метрики accuracy. На рисунках 4 и 5 приведены значения функции потерь sparse_categorical_crossentropy и метрики accuracy модели для 30-ти эпох обучения.

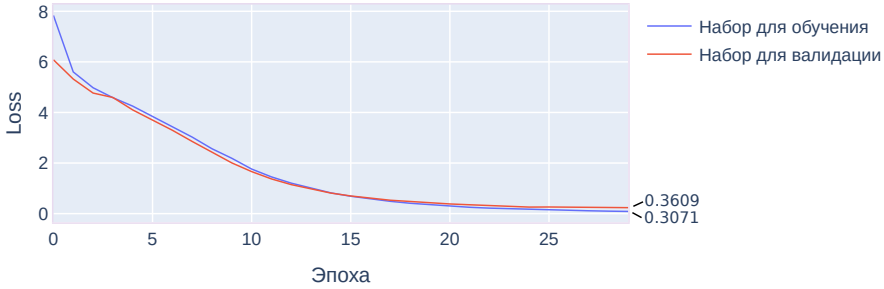

Рисунок 4. Потери модели

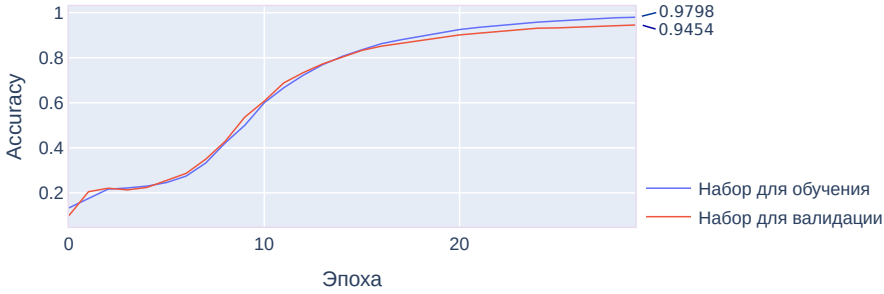

Рисунок 5. Точность модели

Количество эпох найдено экспериментальным путём и является оптимальным. При обучении модели использовался алгоритм стохастической оптимизации rmsprop.

Средние значения метрик BLEU и ROUGE-L, позволяющих оценить точность восстановления неполного текста из набора данных для тестирования, показаны на рисунке 6. Значения из диапазона 0.3-0.4 соответствуют пониманию и приемлемой трансляции текста. Для вычисления значений метрик использовались функции sentence_bleu() и get_scores() из пакетов NLTK и Rouge соответственно. Их вызов осуществлялся после завершения очередной эпохи обучения из callback-функций метода fit().

На рисунке 7 приведены результаты восстановления текста из 10 произвольно сформированных наборов данных по 250–300 предложений

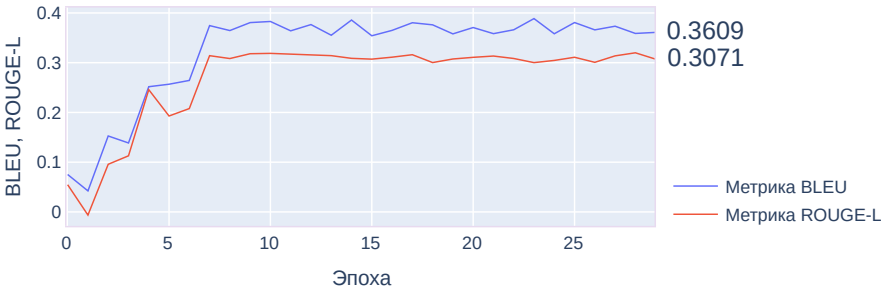

Рисунок 6. Метрики оценки качества восстановления текста

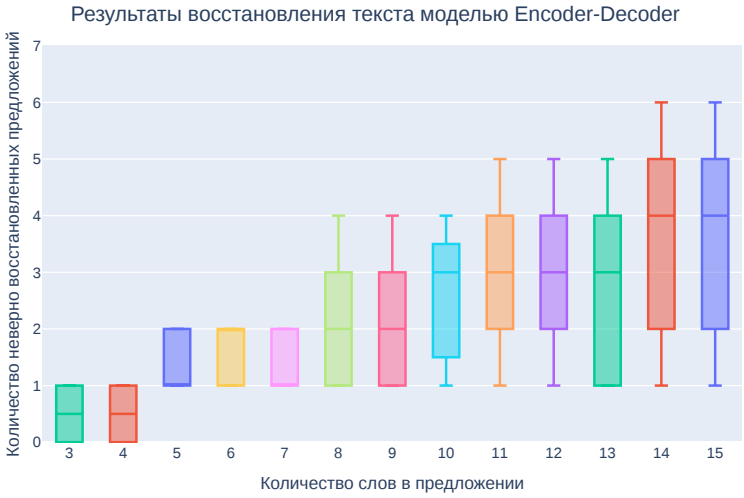

Рисунок 7. Количество неверно восстановленных предложений моделью Encoder-Decoder из 10-ти тестовых наборов данных

для тестирования. Количество неверно восстановленных предложений, состоящих из 11-15 слов, составило от 1 до 6.

5. Формирование и исследование модели трансформера Seq2Seq

Как и для предыдущей модели, формирование и исследование модели с архитектурой Seq2Seq осуществлялось на языке Python с использованием API Keras [**23**–**25**].

Оптимальная структура модели Seq2Seq, полученная в результате экспериментальных исследований, приведена на рисунке 8.

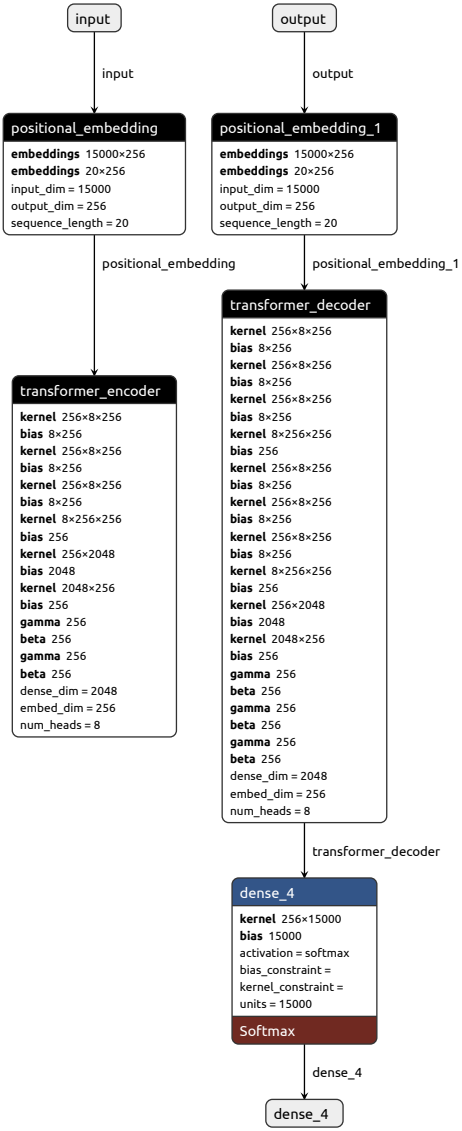

Рисунок 8. Энкодер и декодер модели трансформера Seq2Seq в API Keras для восстановления неполного текста

Модель трансформера состоит из следующих слоёв:

(1) Входной слой (InputLayer) принимает входные данные в виде последовательности слов. Определяет форму и тип входных данных.
(2) Слой позиционных эмбеддингов (Positional Embedding Layer) добавляет информацию о позиции слова в последовательности. Это позволяет модели учитывать порядок слов во входной и выходной последовательностях.
(3) Слои энкодера трансформера (TransformerEncoder Layer), каждый из которых реализует механизм внимания и содержит полносвязные слои. За счёт этого они позволяют моделировать зависимости в последовательностях, извлекать признаки из входных данных и представлять их в оптимальном внутреннем представлении для более сложных вычислений и задач обработки естественного языка
(4) Слои декодера трансформера (TransformerDecoder Layer). Аналогично энкодеру, декодер состоит из нескольких слоев декодера трансформера, которые также включают механизм внимания и полносвязные слои. Декодер генерирует выходную последовательность на основе контекстных представлений энкодера.
(5) Выходной слой (Dense) преобразует предсказанные представления токенов в вероятностное распределение по всем возможным токенам.

Для создания более эффективных и связанных представлений текстовых последовательностей в слоях энкодера и декодера модели с архитектурой Seq2Seq реализован механизм внимания [4]. В таблице 2 приведено описание основных параметров всех слоёв этой модели.

Таблица 2. Слои модели трансформера Seq2Seq

Тип слоя (имя на рисунке 8)	Функция активации	Входной тензор	Выходной тензор
InputLayer (input)	–	[(None, None)]	[(None, None)]
PositionalEmbedding (positional_embedding)	–	(None, None)	(None, None, 256)
TransformerEncoder (transformer_encoder)	relu	(None, None, 256)	(None, None, 256)
InputLayer (output)	–	[(None, None)]	[(None, None)]
PositionalEmbedding (positional_embedding_1)	–	(None, None)	(None, None, 256)
TransformerDecoder (transformer_decoder)	relu	(None, None, 256)	(None, None, 256)
Dense (dense_4)	softmax	(None, None, 256)	(None, None, 15000)

Исследование точности работы модели, по аналогии с предыдущей, заключалось в вычислении значений функции потерь и метрики точности – sparse_categorical_crossentropy и accuracy соответственно для каждой

из 30-ти эпох её обучения, рисунок 9, 10. Количество эпох найдено экспериментальным путём и является оптимальным. При обучении модели использовался алгоритм стохастической оптимизации rmsprop.

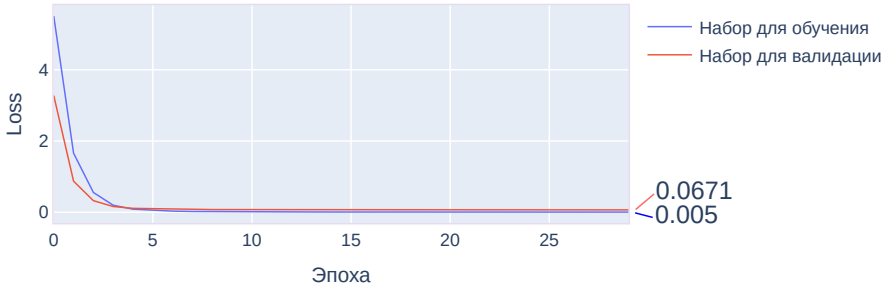

Рисунок 9. Потери модели

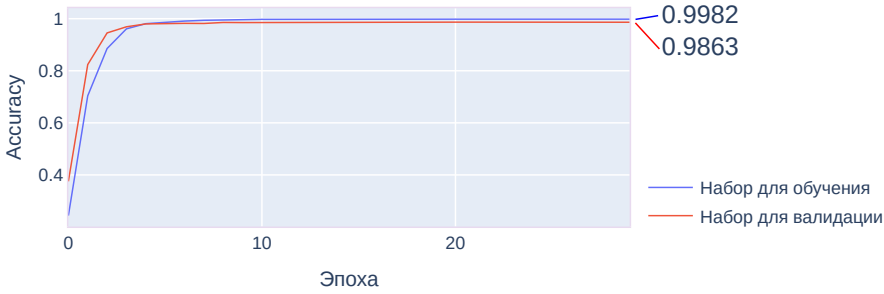

Рисунок 10. Точность модели

Средние значения метрик BLEU и ROUGE-L, позволяющих оценить точность восстановления моделью неполного текста из набора данных для тестирования, показаны на рисунке 11. Значения из диапазона 0.5-0.6 соответствуют высокому качеству трансляции текста.

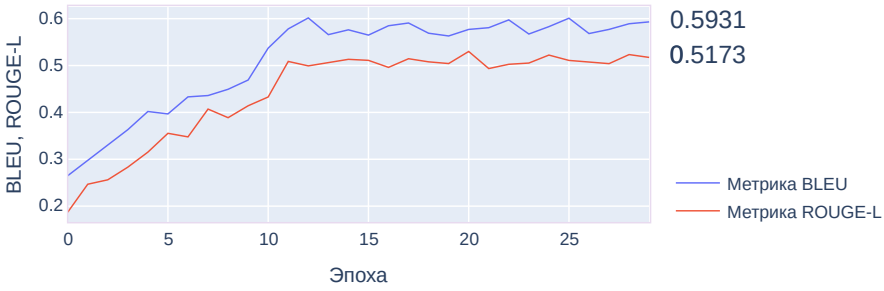

Рисунок 11. Метрики оценки качества восстановления текста

На рисунке 12 приведены результаты восстановления неполного текста из 10-ти произвольно сформированных наборов данных для тестирования. Количество неверно восстановленных предложений меньше, чем для предыдущей модели, и составляет от 1 до 4. Как следствие, отметить вполне закономерный результат – модель трансформера Seq2Seq, за счёт использования механизма внимания, эффективнее модели Encoder-Decoder. Механизм внимания позволяет трансформеру выделять ключевые элементы входных и выходных последовательностей и более эффективно вывялить в них долгосрочные зависимости, что делает модель способной к обучению на сложных задачах генерации (в данном случае – восстановления) текста.

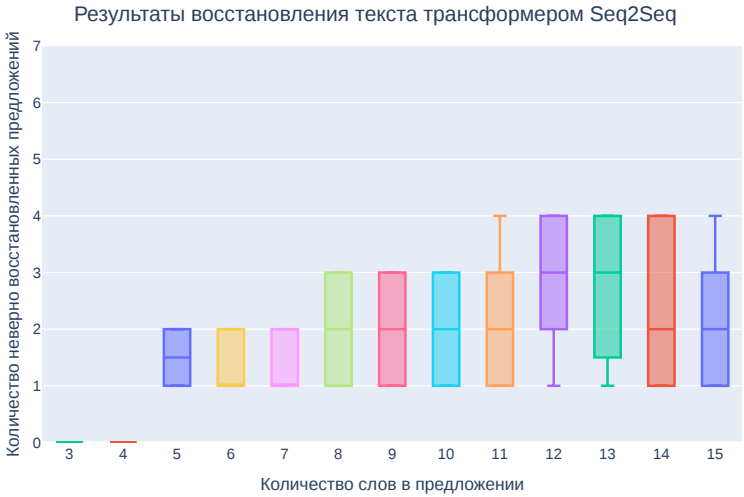

Рисунок 12. Количество неверно восстановленных предложений моделью Seq2Seq из 10-ти тестовых наборов данных

6. Выводы, достоинства и недостатки предложенного подхода

Проведённые в работе исследования показали, что использование DNN-моделей позволяет достаточно эффективно решить поставленную в работе задачу. Приемлемые для решения практической задачи результаты восстановления текста из документов ППК «Роскадастр» объясняются особенностями формирования набора данных – он содержит все пары с неполным и соответствующим ему полным текстами. Помимо этого, неполный текст рассматривался в данной работе как последовательность определённого количества соседних слов, что в значительной мере упрощало процесс его сопоставления с полным текстом. Последовательности из произвольных слов полного текста в набор данных не включались.

Достоинством предложенного подхода является простота моделей и особенностей формирования набора данных для их обучения и валидации.

Недостаток – невозможность восстановления текста по совокупности его произвольных (непоследовательных) слов без существенного усложнения модели, предполагающей анализ контекста предложения. Единственная причина неправильного восстановления текста связана с достаточно редкими случаями (1-3% от общего количества) формирования одинаковых эмбеддингов (векторизация неполного текста и признак области документа, в котором возможно нахождение соответствующего ему полного текста, см. раздел 3). Пример правильного и неправильного восстановления неполного текста с одинаковыми эмбеддингами приведён на рисунке 13.

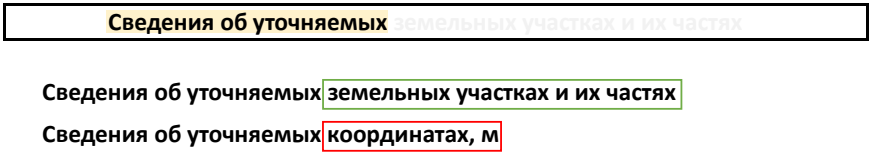

Рисунок 13. Правильно и неправильно восстановленный текст «Сведения об уточняемых» (см. рисунок 1 и 2)

Результаты, полученные в этой работе, планируются к использованию в информационной системе (ИС) ППК «Роскадстр» с целью преобразования отсканированных документов в их текстовые аналоги. Для реализации этого процесса выбрана модель трансформера Seq2Seq, как показавшая лучший по сравнению с моделью Encoder-Decoder результат, таблица 3.

Таблица 3. Значения метрик моделей Encoder-Decoder и Seq2Seq

Метрика loss	Метрика accuracy	Метрика BLEU	Метрика ROUGE-L
Модель Encoder-Decoder			
0.087	0.9798	0.3609	0.3071
Модель трансформера Seq2Seq			
0.005	0.9982	0.5931	0.5173

Метрики, приведённые в этой таблице, имеют отношение только к набору данных, сформированному из документов ППК «Роскадастр». Для наборов данных других предметных областей они могут отличаться.

Список использованных источников

[1] N. C. Sabharwal, A. Agrawal *Hands-on Question Answering Systems with BERT: Applications in Neural Networks and Natural Language Processing.*– Berkeley, CA: Apress.– 2021.– ISBN 978-1-4842-6664-9.– xv+184 pp. ↑94

[2] K. Aitken, V V. Ramasesh, Y. Cao, N. Maheswaranathan *Understanding how encoder-decoder architectures attend.*– 2021.– 24 pp. arXiv 2110.15253~[cs.LG] ↑94

[3] A. Rahali, M. A. Akhloufi *End-to-end transformer-based models in textual-based NLP* // Artificial Intelligence.– 2023.– Vol. **4**.– No. 1.– Pp. 54–110. ↑94

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin *Attention is all you need.*– 2017.– 15 pp. arXiv 1706.03762 ↑94, 105

[5] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu *BLEU: a method for automatic evaluation of machine translation* // *ACL'02 Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (July 7–12, 2002, Philadelphia, Pennsylvania, USA), Stroudsburg: ACL.– 2002.– Pp. 311–318. ↑94

[6] Ch.-Y. Lin *ROUGE: a package for automatic evaluation of summaries* // *Proceedings of the Workshop on Text Summarization Branches Out*, WAS 2004 (July, 2004, Barcelona, Spain).– ACL.– 2004.– 74–81 pp. ↑94

[7] И. В. Винокуров *Использование свёрточной нейронной сети для распознавания элементов текста на отсканированных изображениях плохого качества* // Программные системы: теория и приложения.– 2022.– Т. **13**.– № 3(54).– С. 29–43. ↑96

[8] И. В. Винокуров *Распознавание цифровых последовательностей с использованием свёрточных нейронных сетей* // Программные системы: теория и приложения.– 2023.– Т. **14**.– № 3(58).– С. 3–36 (русс.+англ.). ↑96

[9] Th. Luong, H. Pham, Ch. D. Manning *Effective approaches to attention-based neural machine translation* // *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (17–21 September, 2015, Lisbon, Portugal).– ACL.– 2015.– ISBN 978-1-941643-32-7.– Pp. 1412–1421. ↑97

[10] A. M. Dai, Q. V. Le *Semi-supervised Sequence Learning*, NIPS 2015 (December 7–12, 2015, Montreal, Quebec, Canada), Advances in Neural Information Processing Systems.– Vol. **28**.– Curran Associates, Inc..– 2015.– ISBN 9781510825024.– 9 pp. ↑97

[11] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. N. Dauphin *Convolutional sequence to sequence learning* // *Proceedings of the 34th International Conference on Machine Learning* (6–11 August 2017, International Convention Centre, Sydney, Australia), PMLR.– vol. **70**.– 2017.– Pp. 1243–1252. ↑97

[12] D. Ulyanov, A. Vedaldi, V. Lempitsky *Deep image prior* // International Journal of Computer Vision.– 2020.– Vol. **128**.– No. 7.– Pp. 1867–1888. ↑

[13] K. Hakala, A. Vesanto, N. Miekka, T. Salakoski, F. Ginter *Leveraging text repetitions and denoising autoencoders in OCR post-correction*.– 2019.– 5 pp. arXiv:1906.10907~[cs.CL] ↑97

[14] G. Huang, J. Wang, H. Tang, X. Ye *BERT-based contextual semantic analysis for English preposition error correction* // Journal of Physics: Conference Series.– 2020.– Vol. **1693**.– No. 1.– id. 012115.– 5 pp. ↑97

[15] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu *MASS: masked sequence to sequence pre-training for language generation*, International Conference on Machine Learning (9–15 June 2019, Long Beach, California, USA), PMLR.– vol. **97**.– 2019.– Pp. 5926–5936. arXiv:1905.02450~[cs.CL] ↑97

[16] Sh. Chollampatt, D, T. Hoang, H. T. Ng *Adapting grammatical error correction based on the native language of writers with neural network joint models* // *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2016 (1–4 November, 2016, Austin, Texas, USA).– ACL.– 2016.– ISBN 978-1-945626-25-8.– Pp. 1901–1911. ↑97

[17] A. Graves *Supervised Sequence Labelling with Recurrent Neural Networks*, Studies in Computational Intelligence.– Vol. **385**.– Berlin–Heidelberg: Springer.– 2012.– ISBN 978-3-642-24797-2.– 146 pp. ↑98

[18] T. Ge, X. Zhang, F. Wei, M. Zhou *Automatic grammatical error correction for sequence-to-sequence text generation: an empirical study // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (July 28–August 2, 2019, Florence, Italy).– ACL.– 2019.– ISBN 978-1-950737-48-2.– Pp. 6059–6064. ↑98

[19] X. Zhang, J. Zhao, Y. LeCun *Character-level convolutional networks for text classification.*– 2016.– 9 pp. arXiv 1509.01626~[cs.LG] ↑98

[20] Z. Xie, A. Avati, N. Arivazhagan, D. Jurafsky, A. Ng *Neural language correction with character-based attention.*– 2016.– 10 pp. arXiv 1603.09727~[cs.CL] ↑98

[21] J. Ramirez-Orta, E. Xamena, A. Maguitman, E. Milios, A. Soto *Post-OCR document correction with large ensembles of character sequence-to-sequence models //* Proceedings of the AAAI Conference on Artificial Intelligence.– 2022.– Vol. **36**.– Pp. 11192–11199. ↑98

[22] A. A. Alkhazraji, K. Baheeja, A. M. N. Alzubaidi *Ancient textual restoration using deep neural networks: a literature review // 2023 Al-Sadiq International Conference on Communication and Information Technology*, AICCIT 2023 (04–06 July 2023, Al-Muthana, Iraq).– 2023.– ISBN 9798350341898.– Pp. 64–69. ↑98

[23] F. Chollet *Deep Learning with Python*, 2nd ed..– Manning.– 2021.– ISBN 9781617296864.– 504 pp. ↑99, 103

[24] A. Géron *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed..– Sebastopol: O'Reilly Media.– 2019.– ISBN 978-1-492-03264-9.– 848 pp. ↑99, 103

[25] A. Kapoor, A. Gulli, S. Pal *Deep Learning with TensorFlow and Keras: Build and deploy supervised, unsupervised, deep, and reinforcement learning models*, 3rd ed..– Packt Publishing.– 2022.– ISBN 978-1803232911.– 698 pp. ↑99, 103

Поступила в редакцию 03.03.2024;
одобрена после рецензирования 14.04.2024;
принята к публикации 15.08.2024;
опубликована онлайн 23.09.2024.

Рекомендовал к публикации *д.ф.-м.н. А. М. Елизаров*

Информация об авторе:

Игорь Викторович Винокуров

Кандидат технических наук (PhD), ассоциированный профессор в Финансовом Университете при Правительстве Российской Федерации. Область научных интересов: информационные системы, информационные технологии, технологии обработки данных.

0000-0001-8697-1032

e-mail: *igvvinokurov@fa.ru*

Декларация об отсутствии личной заинтересованности: *благополучие автора не зависит от результатов исследования.*