DOI: 10.14529/cmse230102

A METHOD FOR CREATING STRUCTURAL MODELS OF TEXT DOCUMENTS USING NEURAL NETWORKS

© 2023 D.V. Berezkin, I.A. Kozlov, P.A. Martynyuk, A.M. Panfilkin

Bauman Moscow State Technical University
(st. 2nd Baumanskaya 5/1, Moscow, 105005 Russian Federation)
E-mail: berezkind@bmstu.ru, kozlovilya89@gmail.com,
martapauline@yandex.ru, panfilkinam@student.bmstu.ru
Received: 03.11.2022

The article describes modern neural network BERT-based models and considers their application for Natural Language Processing tasks such as question answering and named entity recognition. The article presents a method for solving the problem of automatically creating structural models of text documents. The proposed method is hybrid and is based on jointly utilizing several NLP models. The method builds a structural model of a document by extracting sentences that correspond to various aspects of the document. Information extraction is performed by using the BERT Question Answering model with questions that are prepared separately for each aspect. The answers are filtered via the BERT Named Entity Recognition model and used to generate the contents of each field of the structural model. The article proposes two algorithms for field content generation: Exclusive answer choosing algorithm and Generalizing answer forming algorithm, that are used for short and voluminous fields respectively. The article also describes the software implementation of the proposed method and discusses the results of experiments conducted to evaluate the quality of the method.

Keywords: information extraction, neural network, named entity recognition, question-answering system.

FOR CITATION

Berezkin D.V., Kozlov I.A., Martynyuk P.A., Panfilkin A.M. A Method for Creating Structural Models of Text Documents Using Neural Networks. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2023. Vol. 12, no. 1. P. 28–45. DOI: 10.14529/cmse230102.

Introduction

Modern information systems accumulate and process huge volumes of heterogeneous data, a significant proportion of which are text documents. Such documents are used as an input for many Natural Language Processing (NLP) tasks that have seen a significant progress in recent years, mostly due to the development of deep learning technologies.

Many NLP tasks require comparing two text documents. Such tasks include text clustering (which requires computing the similarity between two documents in order to determine if they can be put in the same cluster), information retrieval (which involves determining how close a document is to a user's query), plagiarism detection and more. In most cases, comparison of the documents takes into account whole texts of both documents. A typical implementation of such a comparison involves representing the entire document using a vector model (such as Bag of Words, TF-IDF, Word2Vec or other embedding models) and comparing vector models of two documents via various similarity measures such as Cosine Similarity or Word Mover's Distance.

However, in some specific tasks, when comparing documents, only fragments of their texts should be taken into account. Here are some of the possible scenarios in which text documents comparison needs to be performed this way:

1. Comparison of several scientific articles on the same problem in order to determine the most efficient solution. In this case, the articles should be compared in terms of the efficiency of the methods presented.
2. Comparison of several consecutive versions of an official document regulating a certain area (for instance, a national strategy of AI development) in order to track the development of technologies that are used to achieve the goals set in the document.

The task of partial comparison can be easily performed on structured documents when each document is represented by a frame where the fragment of interest is located in a separate field. However, in case of unstructured documents this task is challenging: the respective fragments may be located in different parts of documents and may be worded using different terms. A possible solution to this problem consists of building a structural model of every document and then comparing the models instead of documents themselves. A structural model of a text document is a frame, each element of which corresponds to a certain aspect of the document and each value of an element is a fragment of the text in the document. By "aspect" of the document we mean a semantic component of the text in the document, corresponding to a certain query. For instance, in the case of choosing an article that provides the most efficient solution to a certain problem, the aspect of interest is "Efficiency of the solution" and it can be described by a query "How good is the quality of the proposed method?"

When analyzing scientific articles, possible aspects to extract are the title and authors of the article, the goal and relevance of the research, existing and proposed models and methods. When analyzing national strategies in technical areas, we are interested in other aspects such as the expected time of the strategy's implementation and technologies that are used to achieve the strategy's goals.

A structural model of a text document is created by detecting fragments of the text in the document that corresponds to the aspects of interest. Creation of a structural model can be considered a special case of a general Information Extraction (IE) task which consists of extracting structured information from unstructured natural language text documents. The article considers the solution of this problem using modern neural network technologies.

The article is organized as follows. Section 1 presents the formulation of the problem of creating a structural model of a document as a special case of a general Information Extraction task. Section 2 describes existing approaches to Information Extraction task. Section 3 is devoted to the proposed method and describes its steps and technologies used in each of the steps. In Section 4 we present a software implementation of the proposed method. Section 5 describes experiments conducted to evaluate the quality of the proposed method. Conclusion contains a brief summary of the results obtained in the work and directions for further research.

1. Formulation of the Problem

The general Information Extraction problem can be formulated as determining a set of frame instances $F = \{f_i\}$ and a relation $R_F \subseteq (F \times S \times V)$ based on a set of text documents $D = \{d_i\}$. Each frame instance f_i represents a certain entity extracted from text documents. The relation R_F determines values V of slots S of frame instances F. The slot values V are the fragments extracted from the text documents D.

Various specific Information Extraction tasks fit into this formulation and differ from each other in terms of what objects are represented with frames. Examples of such tasks are the extraction of named entities (names, organizations and geographical locations) [1], addresses [2]

and events [3]. In each of these cases, it is assumed that the analyzed documents D and frame instances F are related in a one-to-many way as multiple entities may be extracted from each document. However, the task that is considered in this article requires that for each document one and only one frame instance is formed — a frame instance that describes the document in a structured way. The slots of this frame instance should correspond to aspects of the document that are of interest to a specific task that is being solved. In this regard, we present a more specific formulation of the problem, taking into account the described requirements.

The task is to determine a set of structural models $M = \{m_i\}$, $i = \overline{1..N_d}$, based on a set of text documents $D = \{d_i\}$, where m_i is a structural model of the document d_i, N_d is the number of documents (as well as the number of structural models). Each structural model m_i is a tuple $m_i = (m_i^j)$, $j = \overline{1..N_a}$, where an element m_i^j is a text string that describes the j-th aspect of the document d_i, and N_a is the number of aspects of interest. Each text string m_i^j is a fragment extracted from the text of the document d_i. Further, we will also use terms “card” and “fields” to denote the structural model and its elements respectively.

2. Related Work

Traditionally, Information Extraction tasks are solved mainly using two approaches: rule-based and probabilistic. However, due to the rapid development of neural networks in recent years, they have found application in solving various problems of text analysis, including the problem of Information Extraction. In this section, we will consider both traditional approaches and the neural network approach.

2.1. Rule-based Approach

Rule-based Information Extraction methods use extraction rules written in a formal language. An extraction rule imposes a set of restrictions on the analyzed text fragment. These restrictions may apply to orthographic, morphological, syntactic and semantic features of separate words, as well as to relations between them. If the text fragment meets the rule's restrictions, it is concluded that this fragment contains the sought-for entity. In this case, a new frame instance is created that represents the extracted entity. Its slots are initialized with some elements of the text fragment.

Depending on the formal language that is used for writing rules, rule-based Information Extraction methods can be divided into two categories: propositional and relational [4]. Propositional methods use rules that are written in the language of zero order (propositional) logic. Expressions in such a language can only include attributes of words and phrases. The most common attributes are morphological features of words, syntactic roles of words in a sentence, and semantic classes. Relational methods use rules that are written in the language of first order logic. In addition to attributes of words and phrases, such rules can describe relations between them. The most common types of such relations are syntactic and order relations that specify the syntactic structure of a sentence and the order of phrases within a sentence respectively.

The rules can be written by an expert or generated automatically based on a set of training examples [5]. Training examples are manually tagged by experts and then are used by the Information Extraction system to infer extraction rules by generalizing restrictions during the learning process.

A rule-based approach CLIEL proposed in [6] consists of two stages of processing: organizing the text in an accessible form and subsequent extraction of information. The basis of the

mechanism is the recognition of the document layout and the use of a set of grammatical rules to extract information from commercial law documents. An approach proposed in the paper [7] also takes into account the structure of the document and uses rules to extract information. First, the structure of the document is revealed to determine what information should be extracted from its individual parts. In order to search for the relevant part of the text to extract specific information, a compact lexical dictionary is used. Second, the text is normalized, tokenized, tagged using POS Tagger, and information is extracted using templates.

2.2. Probabilistic Approach

Probabilistic Information Extraction methods are based on the construction of probabilistic models that include observable and hidden variables. Observable variables X correspond to various features of the analyzed text fragment. Hidden variables Y match elements of the text fragment to the slots of the frame that represents the entity that is being extracted. When analyzing a certain text fragment described by features x, the values of hidden variables y are determined by maximizing the conditional probability $\mathbf{P}(Y = y|X = x)$. If the value of this probability exceeds a certain threshold, a new frame instance is created and its slots are filled with elements of the text fragment in accordance with y. The probabilistic models are trained by estimating their parameters using the maximum likelihood method.

Probabilistic Information Extraction methods can use generative or discriminative models. Generative models are based on calculating the joint probability $\mathbf{P}(Y = y, X = x)$ that is then used to determine the probability $\mathbf{P}(Y = y|X = x)$. They include, among others, Naive Bayes Classifier [8] and Hidden Markov Model [9]. Discriminative models allow to directly determine the desired conditional probability $\mathbf{P}(Y = y|X = x)$. These include Conditional Random Fields [10].

The paper [11] analyzes several probabilistic models that have proven to be particularly useful for various tasks of extracting meaning from natural language texts. Most prominent among them are Hidden Markov models (HMMs), stochastic context-free grammars (SCFG), and maximal entropy (ME).

2.3. Neural Network Approach

The use of traditional methods requires a large amount of linguistic resources: tagged corpus of texts, dictionaries, thesauri. In order to take into account all the ways in which the aspects of interest can be described in documents, it is necessary to prepare a large number of rules, which requires a lot of expert work and a high level of knowledge in the subject area. These problems can be avoided via the usage of methods based on modern neural network technologies due to the ability of neural networks to independently determine the feature space when processing the training corpus of text documents. This technology is called representation learning. It aims to automatically obtain informative representations of objects from raw data. Hence, preliminary training of modern language models does not require a detailed tagging of text elements with features (morphological, syntactic, semantic, and others). As a result of training, the neural network is able to represent every word with a vector of numbers (so-called “word embedding”), each of which is a value of some feature selected by the network. Unlike traditional vector models (such as “bag of words” and TF-IDF), embeddings reflect the semantics of words, and also allow to evaluate their contextual proximity. Deep learning models trained to form word embeddings are called pre-trained. The pre-trained models are universal and are used to solve NLP problems in various subject areas. To solve a specific NLP problem, a pre-trained model needs fine-tuning,

which requires significantly less data than pre-training. At the moment, ready-to-use datasets are available for fine-tuning and testing neural network models. Datasets for NLP tasks are usually taken from the collections of the GLUE benchmark [12].

One of the latest and most ambitious developments in the field of neural network language models is the BERT model, released in 2018 by Google [13]. Due to the architectural features of this model, BERT is able to take into account the bidirectional context of words when pre-training representations, which gives it an advantage over other neural network language models such as Word2Vec and GloVe [14]. Thus, the model creates different embeddings for homonymous words, which allows avoiding false interpretations of words in further work with text documents.

BERT model is based on the Transformer architecture and uses the Self-Attention mechanism, which makes it easy to adapt the model for solving specific NLP tasks by fine-tuning it on task-relevant input and output data [13]. Fine-tuned BERT-based neural network models have been used for solving various NLP tasks including the Information Extraction problem. For instance, the DeepPavlov open source library contains special purpose BERT models configured to solve the problem of Named Entity Recognition (NER) [15]. Also, the DeepPavlov library contains a BERT model fine-tuned for extracting relations between objects. Named Entity Recognition and relation extraction are subtasks of the Information Extraction problem that have been widely studied by researchers from the perspective of possible use of modern neural network models [16]. However, the existing methods that have been proposed in this area are highly specialized and cannot be directly applied to solve the task of building structural models of text documents.

There has been a tendency to solve various NLP problems by converting them to the Question-Answering (Q&A) task [17]. For example, the paper [18] proposes using a Q&A model for extracting named entities. Paper [19] describes multi-turn Q&A method for relation extracting using question templates. Authors use BERT model as a backbone for the Q&A framework. Paper [20] implements information extraction system based on the functioning of a Q&A model proposed by authors and named QA4IE. Since Q&A-based approach has been proven effective for solving IE tasks, we decided to use it as a basis for our method for creating structural models of text documents.

3. The Proposed Method for Creating Structural Models of Text Documents

In order to build structural models of documents we use BERT based neural network models. One of them is the Q&A model. It is fine-tuned on the SQuAD (Stanford Question Answering Dataset) dataset, which consists of questions based on Wikipedia articles, where the answer to each question is a text fragment of an article [21]. Another model that we use is the NER model fine-tuned on the OntoNotes dataset, containing marked-up data from various sources (web blogs, telephone conversations, news feeds) and supporting 18 entity categories [22]. Both Q&A and NER models are open source and ready to use: BERT Q&A is provided in the official Google Research github repository [23] and BERT NER can be obtained from the DeepPavlov open source library [24]. In addition, we use the Sentence-BERT [25, 26] model that generates sentence embeddings, in contrast to the usual BERT embeddings that are formed for tokens — words or parts of words contained in the model dictionary. All models are already pre-trained and fine-tuned, they do not require additional training. All of them are generic English language

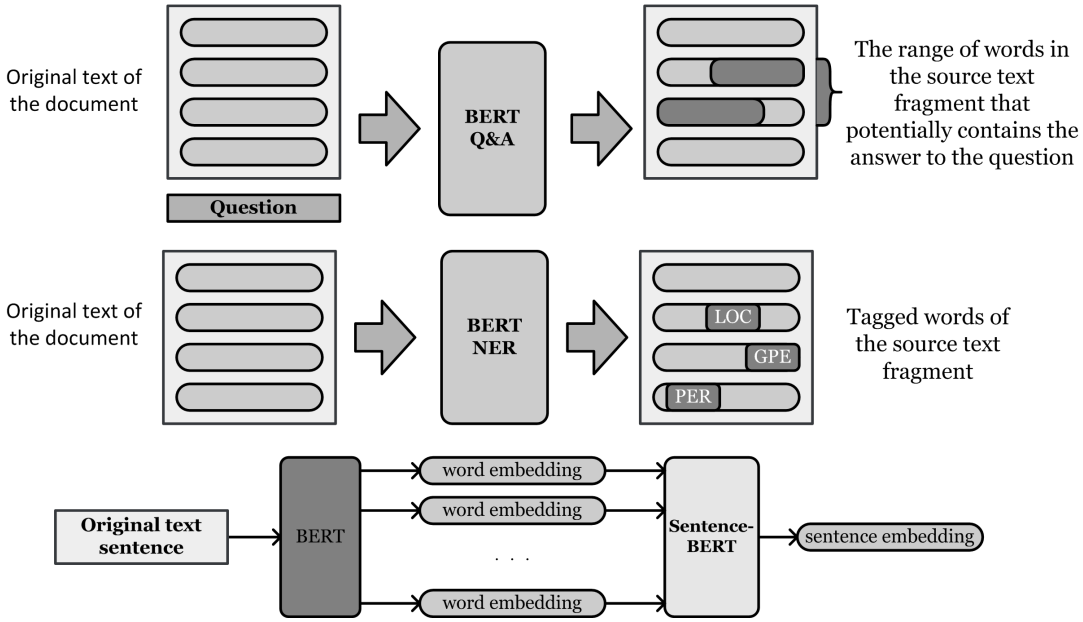


Fig. 1. The concept of application of the used BERT models

models not tailored for any specific domain, whereupon they can be used to process documents of various types and topics. As a neural network framework PyTorch framework is used [27].

The concept of application of the used BERT models is shown in Fig. 1.

The BERT Q&A model is used to search for pieces of the text in the document that contain information about certain aspects of the structural model. Since the structural model is presented as a card with fields, the task of finding information about an aspect of the model becomes the task of filling in the card field dedicated to this aspect. In this case, the desired content of the card field can be searched as an answer to some given question that describes the aspect.

As an input sequence, the BERT Q&A model uses the question text and the text to search for the answer in token format. Due to the specific architecture of the BERT Q&A model, the size of the input sequence of tokens should not exceed 512 elements. This imposes restrictions on the length of an input text to search for the answer. Since the model cannot scan the entire text of the document in search of the answer at once, it is necessary to break the text into fragments of an appropriate length. In order to prepare the fragments, the text is preliminarily cleared of formatting characters and divided into sentences. The text fragment is formed by sequentially adding consecutive sentences until the maximum possible number of tokens is reached.

Sequential forming of the fragments leads to the following problem. A potential answer to the question can be voluminous, consisting of multiple sentences. In this case, the beginning of a potential answer may be in one fragment and the ending in the other. There is a risk that the BERT Q&A model will be able to identify a potential answer from the first fragment, but not from the other one. In order to reduce the risk of receiving an incomplete answer by the system, it was decided to allocate text fragments with an overlap on each other.

Not all of the formed fragments actually contain answers to the question. Sometimes the desired answer (for example, the name of the author of the document) appears in the text only 1 or 2 times. In this case, it can be present in just one text fragment. In order to reduce the number of text fragments to search for the answer, we filter the fragments using relevant words that describe the aspect. Further analysis will only consider relevant fragments (that is, fragments that contain the relevant words).

The questions that are used to prepare the input of the Q&A model represent the meaning of the aspect. Using a set of questions instead of just one question can increase the system's chances to correctly identify the data we are looking for. Thus, by dividing the text into fragments for each question from the set, filtering out the relevant ones, using the BERT Q&A model and combining the results obtained for different questions, it is possible to get an array of answers that contain information about the aspect.

To make the method adjustable for various domains, the user should be able to define and customize all of the fields of the document card. This imposes the need to provide a universal and flexible approach to setting up a field content search, which implies choosing relevant word sets and questions. However, words in the set should not be too general (in this case, filtering may be useless) or only specialized (then there may not be relevant text fragments at all). Questions should be formulated based on the following principles. First, they should be as short as possible (the size of text fragments depends on the length of the question). Second, they should be specific. Empirically, it was found that the best result was obtained using questions *Who*, *What*, *When*, *Where*. Third, all questions related to the same field should receive the same answer according to the proposed methodology.

In some cases, we know exactly which type of named entities should be contained in the answers. For example, when searching for the name of the author of a document, the answers must explicitly contain an entity of the PER (person's name) category. For additional filtering (in order to remove the answers that do not contain a desired entity), it is proposed to use the BERT NER model. In order to apply this filtering, it is necessary to determine a list of required named entities for every aspect.

Document card fields can be either short (for example, document title, author's name) or voluminous (for example, a list of modern technologies mentioned in the document). In this regard, it is necessary to use different strategies for processing the received array of answers in order to form the final content of the card field.

Taking into account the chosen methodology for using the BERT Q&A model and BERT NER model to form the content of card fields, each field (that is, each aspect of the structural model) should be provided with the following data:

- a set of questions, the answers to which should be included in the content of a field;
- a set of words which are thematically relevant to the content of the card field;
- a set of categories of named entities (named entities tags);
- a field type mark (short or voluminous).

By determining the set of fields and providing each of them with such data, the proposed method can be adjusted to different types of documents and various topics.

The full scheme of the proposed concept of using the BERT models to form the content of the document card field is shown in Fig. 2.

On Stage 1 the data is prepared for processing. The source text of the document, extracted from the PDF file, is cleaned of formatting characters and split into sentences.

Stage 2 consists of breaking the source text into text fragments for search and forming input sequences for the BERT Q&A model. The generated fragments are filtered using the relevant words. If the fragment does not meet the minimum requirements of relevant words, it is discarded and cannot be used for further analysis.

At Stage 3 the BERT Q&A model is applied to those text fragments that contain the relevant words. As a result, at Stage 4, an array of answers is received.

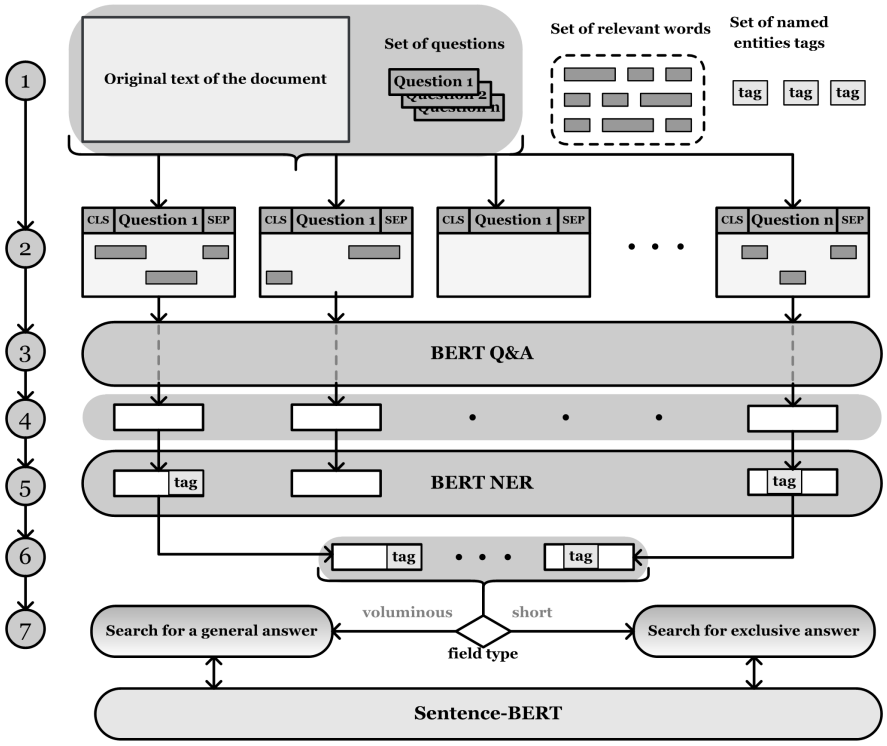


Fig. 2. The proposed concept of using the BERT models to form the content of the document card field

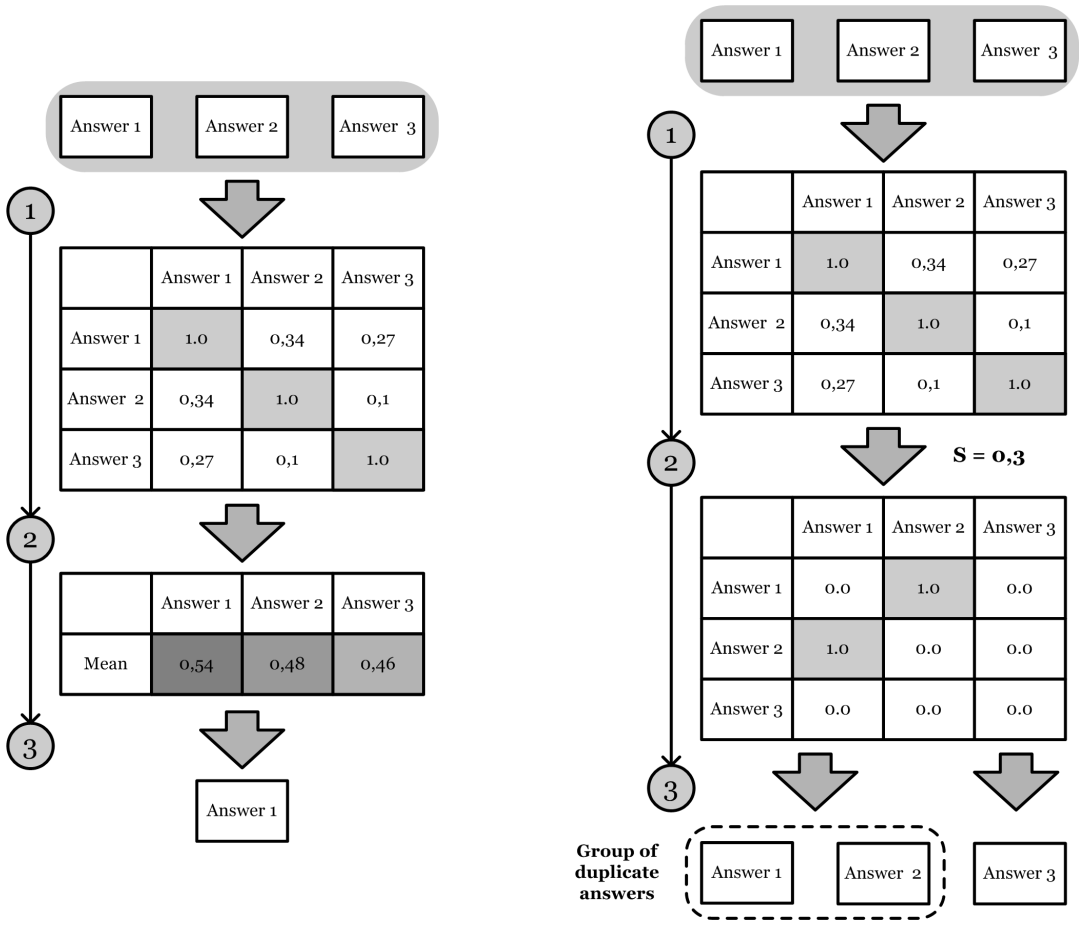
At Stage 5, the answers from the received array are fed to the BERT NER model in order to search them for named entities of the specified categories. Answers in which entities are not found are filtered out. As a result, at Stage 6, an array of answers containing the words of the required categories is obtained.

Stage 7 consists of the formation of the final content of the card field. One of two algorithms is applied to the array of answers, depending on the type of the field being processed. In the case of a short field, an exclusive answer choosing algorithm is applied, and in the case of a voluminous field, a generalizing answer forming algorithm is applied. The visualization of the proposed algorithms is shown in Fig. 3.

The algorithm for choosing an exclusive answer (Fig. 3a) is implemented as follows. A semantic similarity matrix is formed for the array of answers, where each cell of the matrix contains the result of comparing two corresponding answers. The comparison is performed by calculating the cosine measure of similarity between the vector representations of the answers that are obtained via Sentence-BERT model. For each of the answers, the average value of its measure of similarity with other answers in the array is calculated. The answer with the highest average value is chosen as an exclusive answer and used to fill the content of the card field.

The generalizing answer forming algorithm includes the search for duplicate answers in the array (Fig. 3b). For answers, a similarity matrix is constructed in a similar way. Then, the numerical values in the cells are converted according to the rule:

$$Sim'_{ij} = \begin{cases} 0, & \text{if } Sim_{ij} < S \text{ or } i = j, \\ 1, & \text{if } Sim_{ij} > S. \end{cases} \tag{1}$$



(a) Exclusive answer choosing algorithm (b) Generalizing answer forming algorithm

Fig. 3. The visualization of the proposed algorithms

Here Sim'_{ij} is the new content of the cell, Sim_{ij} is the initial value, S is the similarity threshold value set by the user.

Cells containing the ones are grouped, resulting in clusters of duplicate answers. From each cluster, the best answer is chosen using the algorithm for choosing an exclusive answer that has been described earlier. After that, for the final array of unique answers, which are words or phrases, the original full sentences are extracted from the original text of the document. The concatenation of these sentences is considered as the generalizing answer and used to fill the content of the card field.

4. Software Implementation of the Proposed Method

In order to test the proposed method we implemented a system for building structural models of text documents. The system consists of three main blocks shown in Fig. 4: the interface block, the data processing block and the data storage block.

The interface block serves as the means for the user to interact with the system. Through this block, the user can upload files for further processing, launch, manage and monitor data processing, and also view the constructed structural models of documents. The block is implemented as a Docker container with the apache2 web server running in it, which processes the requested PHP pages.

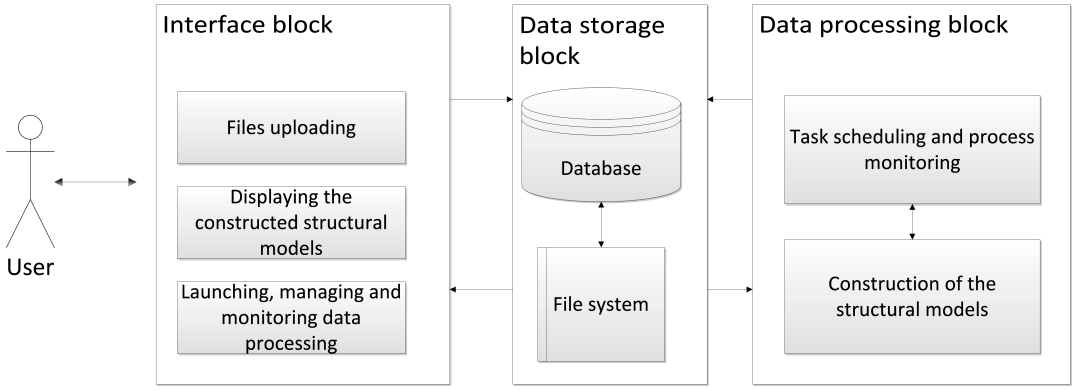


Fig. 4. The block scheme of the system for building structural models of text documents

The data processing block is the core of the system. This block is also implemented as a separate Docker container with a task scheduler running in it, which periodically checks the database for new tasks and starts their execution, in particular, the construction of structural models of documents. Task scheduling and data processing functions are implemented as Python scripts.

The data storage block serves as a link between the interface block and the data processing block. Documents uploaded by the user enter the file system, and information about the uploaded files is recorded in the database. The task scheduler uses the database to store data about scheduled and running processes. Structural models constructed by the data processing block are also stored in the database.

5. Experimental Results

We conducted a series of experiments using several sets of documents. The first set was compiled from scientific articles related to artificial intelligence and data processing that were presented in recent years at ACM conferences such as ICMLC (machine learning and computing), IR (information retrieval) and WSDM (web search and data mining). The second set was compiled from archives of AIAA Journal of Aircraft and AIAA Journal of Spacecraft and Rockets (about 16,000 articles in total). The articles from the sets were processed using the system and structural models were generated for each of them.

Before processing the articles, we set up the system by preparing the initial data for 9 fields: 4 short and 5 voluminous. Short fields described the general features of the article (title and authors) and the conference where it was presented (name and date). Voluminous fields described various aspects of the article's content such as the goal and relevance of the research, existing and proposed models and methods, performance and quality of the proposed methods.

In order to assess the quality of the proposed method, we prepared reference structured models for the articles by manually extracting fragments of the articles that described the desired aspects. Then, models generated by the system were compared to models manually prepared by experts. As the result of the models comparison, the degree of semantic similarity was calculated for each field. The similarity score takes values from 0 to 1 and shows how well the field value generated by the system corresponds to the value determined by the expert. The similarity score for short fields is determined using a Sentence BERT model and a measure of cosine similarity. The similarity score for voluminous fields is calculated via a modified Jaccard's binary similarity

measure as the percentage of similar sentences in the compared texts relative to the total number of unique sentences in both texts.

Table 1 demonstrates the result of comparison of models generated by the system and manually prepared by experts. By "good match" we denote a case in which the card field generated by the system allows to largely learn the respective aspect of the article without reading the article itself (which empirically corresponds to the similarity score exceeding 0.7). By "partial match" we denote a case where at least part of the information related to the aspect is present in the card field generated by the system (which corresponds to the similarity score exceeding 0.3). The last column of the table ("At least partial matches") contains the fraction of articles for which the content of a card field generated by the system fully or partially matches the content prepared by an expert. We use this value to assess the quality of the structured models built by the system.

Table 1. The result of comparison of models generated by the system and prepared by experts

Field name	Type	Good matches	Partial matches	At least partial matches
Title of the article	short	21%	31%	52%
Authors of the article	short	47%	0	47%
Conference name	short	68%	0	68%
Conference date	short	68%	0	68%
The goal of the research	voluminous	31%	21%	52%
Relevance of the research	voluminous	37%	37%	74%
Existing models/methods	voluminous	26%	37%	63%
Proposed models/methods	voluminous	42%	32%	74%
Performance and quality	voluminous	53%	21%	74%

The experiment demonstrated relatively high quality for voluminous fields. The "Relevance of the research", "Proposed models and methods" and "Performance and quality" fields were at least partially correctly recognized for 74% of articles, the "Existing models and methods" field — for 63% of articles. Further improvement of quality may be achieved by adjusting sets of questions and words for aspects. In general, the results of the experiment demonstrated the ability of the system to extract fragments of interest from various unstructured natural language texts in a uniform manner.

Aspects that had the lowest quality of recognition were: "Title of the article" and "Authors of the article" (52% and 47% respectively). This is because the proposed method extracts information based on the proximity of the meaning of the extracted fragment and its context to the question. However, the title and the list of authors have no context: they are located separately in a special place of the article. Also, the content of these fields is unique for each article and cannot be described by a set of questions. Therefore, to extract these aspects, it is necessary to use other methods based on utilizing information about the structure of the article.

In order to check the versatility of the developed method, we also carried out experiments using another type of documents, namely national strategies in technical areas such as Artificial Intelligence. Experiments showed the ability of the system to extract aspects such as goals of the strategy, organizations responsible for implementation of the strategy and technologies that are used to achieve the strategy's goals.

The experiments were carried out on sets of documents of various sizes (from tens to tens of thousands of documents). They confirmed the scalability of the system.

Conclusion

In this work, we proposed an Information Extraction method that forms structural models of text documents. A structural model of a document is a card, each field of which contains text that describes a certain aspect of the document. The proposed method is based on applying a Question Answering neural network model to fragments of the text in the document in order to generate answers that potentially describe the target aspect. The fragments are filtered using sets of relevant words and lists of required named entities. The array of answers generated by the Q&A model is used to form a card field of a document. The formation of the field is performed using one of two algorithms depending on the type of the field. In the case of a short field, an exclusive answer choosing algorithm is applied, and in the case of a voluminous field, a generalizing answer forming algorithm is applied.

We presented a system for building structural models of documents that implements the proposed method. The system allows the user to upload documents, manage their processing and view the constructed structural models. We analyzed the quality of the proposed method via an experiment conducted on a set of scientific articles.

We plan to further develop the proposed method in order to overcome the discovered problems and increase the quality of recognition. We plan to conduct additional experiments in order to evaluate the contribution of using NER to the quality of information extraction, and also to compare the proposed method with other approaches such as rule-based and probabilistic approach.

This paper is a part of the research work carried out within the Bauman Deep Analytics project of the Priority 2030 program.

References

1. Mansouri A., Affendey L.S., Mamat A. Named entity recognition approaches. International Journal of Computer Science and Network Security. 2008. Vol. 8, no. 2. P. 339–344.

2. Brown D.E., Liu X. Extracting Addresses from News Reports Using Conditional Random Fields. Proceedings of the 15th IEEE International Conference on Machine Learning and Applications, ICMLA, Anaheim, California, USA, December 18–20, 2016. IEEE, 2016. P. 791–795. DOI: 10.1109/ICMLA.2016.0141.

3. Benson E., Haghighi A., Barzilay R. Event discovery in social media feeds. Association for Computational Linguistics: Human Language Technologies, 49th Annual Meeting, HLT '11, Portland, Oregon, USA, June 19–24, 2011. Proceedings. Vol. 1. Association for Computational Linguistics, 2011. P. 389–398.

4. Turmo J., Ageno A., Catala N. Adaptive information extraction. ACM Computing Surveys. 2006. Vol. 38, no. 2. P. 1–47. DOI: 10.1145/1132956/1132957.

5. Chai J.Y., Biermann A.W., Guinn C.I. Two dimensional generalization in information extraction. Proceedings of the Sixteenth National Conference on Artificial Intelligence, AAAI-99, Orlando, Florida, USA, July 18–22, 1999. American Association for Artificial Intelligence, 1999. P. 431–438.

6. García-Constantino M., Atkinson K., Bollegala D., *et al.* CLIEL: Context-based information extraction from commercial law documents. Proceedings of the 16th International Conference on Artificial Intelligence and Law, ICAIL'17, London, UK, June 12–16, 2017. Association for Computing Machinery, 2017. P. 79–87. DOI: 10.1145/3086512.3086520.

7. Kadhim K.J., Sadiq A.T., Abdulah H.S. Unsupervised-Based Information Extraction from Unstructured Arabic Legal Documents. Opción: Revista de Ciencias Humanas y Sociales. 2019. Vol. 35, no. 20. P. 1097–1117.

8. Freitag D. Machine learning for information extraction in informal domains. Machine learning. 2000. Vol. 39, no. 2. P. 169–202. DOI: 10.1023/A:1007601113994.

9. Borkar V., Deshmukh K., Sarawagi S. Automatic segmentation of text into structured records. Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, SIGMOD'01, Santa Barbara, California, USA, May 21–24, 2001. Association for Computing Machinery, 2001. P. 175–186. DOI: 10.1145/375663.375682.

10. McCallum A. Efficiently inducing features of conditional random fields. Uncertainty in Artificial Intelligence, Proceedings of the Nineteenth Conference, UAI03, Acapulco, Mexico, August 7–10, 2003. Morgan Kaufmann, 2003. P. 403–410.

11. Feldman R., Sanger J. Probabilistic Models for Information Extraction. The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, 2006. P. 131–145.

12. Wang A., Singh A., Michael J., *et al.* GLUE: a multi-task benchmark and analysis platform for natural language understanding. Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, Louisiana, USA, May 6–9, 2019. P. 1–20. DOI: 10.18653/v1/w18-5446.

13. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, Minnesota, USA, June 2–7, 2019. Vol. 1: Long and Short Papers. Association for Computational Linguistics, 2019. P. 4171–4186. DOI: 10.18653/v1/n19-1423.

14. Pennington J., Socher R., Manning C.D. Glove: Global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Doha, Qatar, October 25–29, 2014. Association for Computational Linguistics, 2014. P. 1532–1543. DOI: 10.3115/v1/d14-1162.

15. Burtsev M., Seliverstov A., Airapetyan R., *et al.* DeepPavlov: Open-Source Library for Dialogue Systems. Association for Computational Linguistics-System Demonstrations, Proceedings of the 56th Annual Meeting, Melbourne, Australia, July 15–20, 2018. Association for Computational Linguistics, 2018. P. 122–127. DOI: 10.18653/v1/p18-4021.

16. Xue K., Zhou Y., Ma Z., *et al.* Fine-tuning BERT for joint entity and relation extraction in Chinese medical text. Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM, San Diego, California, USA, November 18–21, 2019. IEEE, 2019. P. 892–897. DOI: 10.1109/bibm47256.2019.8983370.

17. Wang Q., Yang L., Kanagal B., *et al.* Learning to Extract Attribute Value from Product via Question Answering: A Multi-task Approach. Proceedings of the 26th ACM

SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD'20, USA, August 23–27, 2020. Association for Computing Machinery, 2020. P. 47–55. DOI: 10.1145/3394486.3403047.

18. Banerjee P., Pal K.K., Devarakonda M.V., Baral C. Biomedical Named Entity Recognition via Knowledge Guidance and Question Answering. ACM Transactions on Computing for Healthcare. 2021. Vol. 2, no. 4. P. 1–24. DOI: 10.1145/3465221.
19. Li X., Yin F., Sun Z., *et al.* Entity-Relation Extraction as Multi-Turn Question Answering. Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 – August 2, 2019. Vol. 1: Long Papers. Association for Computational Linguistics, 2019. P. 1340–1350. DOI: 10.18653/v1/p19-1129.
20. Qiu L., Ru D., Long Q., Zhang W., Yu Y. QA4IE: A Question Answering Based Framework for Information Extraction. Proceedings of the 17th International Semantic Web Conference, ISWC 2018, Monterey, California, USA, October 8–12, 2018. Vol. 11136 / ed. by D. Vrandečić, K. Bontcheva, M.C. Suárez-Figueroa, *et al.* Springer, 2018. P. 198–216. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-00671-6_12.
21. Rajpurkar P., Jia R., Liang P. Know What You Don't Know: Unanswerable Questions for SQuAD. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018. Vol. 2: Short Papers. Association for Computational Linguistics, 2018. P. 784–789. DOI: 10.18653/v1/p18-2124.
22. Weischedel R., Hovy E., Marcus R., *et al.* OntoNotes: A large training corpus for enhanced processing. Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation / ed. by J. Olive, C. Christianson, J. McCary. Springer, 2011.
23. Google Research Github Account. TensorFlow code and pre-trained models for BERT. URL: `https://github.com/google-research/bert` (accessed: 31.10.2022).
24. DeepPavlov lab Github Account. An open source library for deep learning end to end dialog systems and chatbots. URL: `https://github.com/deeppavlov/DeepPavlov` (accessed: 31.10.2022).
25. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Hong Kong, China, November 3–7, 2019. Association for Computational Linguistics, 2019. P. 3982–3992. DOI: 10.18653/v1/D19-1410.
26. Ubiquitous Knowledge Processing Lab Github Account. Multilingual Sentence & Image Embeddings with BERT. URL: `https://github.com/UKPLab/sentence-transformers` (accessed: 31.10.2022).
27. An open source machine learning framework PyTorch. URL: `https://pytorch.org/` (accessed: 31.10.2022).

УДК 004.89 DOI: 10.14529/cmse230102

МЕТОД СОЗДАНИЯ СТРУКТУРНЫХ МОДЕЛЕЙ ТЕКСТОВЫХ ДОКУМЕНТОВ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

© 2023 Д.В. Березкин, И.А. Козлов, П.А. Мартынюк, А.М. Панфилкин

Московский государственный технический университет имени Н.Э. Баумана

(105005 Москва, ул. 2-я Бауманская, д. 5, стр. 1)

E-mail: berezkind@bmstu.ru, kozlovilya89@gmail.com,

martapauline@yandex.ru, panfilkinam@student.bmstu.ru

Поступила в редакцию: 03.11.2022

В статье описываются современные нейросетевые модели на основе BERT и рассматривается их применение для задач обработки естественного языка (NLP), таких как ответы на вопросы и распознавание именованных сущностей. В статье представлен метод решения задачи автоматического создания структурных моделей текстовых документов. Предлагаемый метод является гибридным и основан на совместном использовании нескольких моделей NLP. Метод строит структурную модель документа, извлекая предложения, соответствующие различным аспектам документа. Извлечение информации осуществляется с использованием вопросно-ответной модели BERT с вопросами, подготовленными отдельно для каждого аспекта. Ответы фильтруются с помощью модели распознавания именованных сущностей BERT и используются для формирования содержимого каждого поля структурной модели. В статье предложены два алгоритма формирования содержимого поля — алгоритм выбора исключающего ответа и алгоритм формирования обобщающего ответа, которые используются для коротких и объемных полей соответственно. В статье также описывается программная реализация предлагаемого метода и обсуждаются результаты экспериментов, проведенных для оценки качества метода.

Ключевые слова: извлечение информации, нейронная сеть, распознавание именованных сущностей, вопросно-ответная система.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Berezkin D.V., Kozlov I.A., Martynyuk P.A., Panfilkin A.M. A Method for Creating Structural Models of Text Documents Using Neural Networks // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2023. Т. 12, № 1. С. 28–45. DOI: 10.14529/cmse230102.

Литература

1. Mansouri A., Affendey L.S., Mamat A. Named entity recognition approaches // International Journal of Computer Science and Network Security. 2008. Vol. 8, no. 2. P. 339–344

2. Brown D.E., Liu X. Extracting Addresses from News Reports Using Conditional Random Fields // Proceedings of the 15th IEEE International Conference on Machine Learning and Applications, ICMLA, Anaheim, California, USA, December 18–20, 2016. IEEE, 2016. P. 791–795. DOI: 10.1109/ICMLA.2016.0141.

3. Benson E., Haghighi A., Barzilay R. Event discovery in social media feeds // Association for Computational Linguistics: Human Language Technologies, 49th Annual Meeting, HLT '11, Portland, Oregon, USA, June 19–24, 2011. Proceedings. Vol. 1. Association for Computational Linguistics, 2011. P. 389–398.

4. Turmo J., Ageno A., Catala N. Adaptive information extraction // ACM Computing Surveys. 2006. Vol. 38, no. 2. P. 1–47. DOI: 10.1145/1132956/1132957.

5. Chai J.Y., Biermann A.W., Guinn C.I. Two dimensional generalization in information extraction // Proceedings of the Sixteenth National Conference on Artificial Intelligence, AAAI-99, Orlando, Florida, USA, July 18–22, 1999. American Association for Artificial Intelligence, 1999. P. 431–438.

6. García-Constantino M., Atkinson K., Bollegala D., *et al.* CLIEL: Context-based information extraction from commercial law documents // Proceedings of the 16th International Conference on Artificial Intelligence and Law, ICAIL'17, London, UK, June 12–16, 2017. Association for Computing Machinery, 2017. P. 79–87. DOI: 10.1145/3086512.3086520.

7. Kadhim K.J., Sadiq A.T., Abdulah H.S. Unsupervised-Based Information Extraction from Unstructured Arabic Legal Documents // Opción: Revista de Ciencias Humanas y Sociales. 2019. Vol. 35, no. 20. P. 1097–1117.

8. Freitag D. Machine learning for information extraction in informal domains // Machine learning. 2000. Vol. 39, no. 2. P. 169–202. DOI: 10.1023/A:1007601113994.

9. Borkar V., Deshmukh K., Sarawagi S. Automatic segmentation of text into structured records // Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, SIGMOD'01, Santa Barbara, California, USA, May 21–24, 2001. Association for Computing Machinery, 2001. P. 175–186. DOI: 10.1145/375663.375682.

10. McCallum A. Efficiently inducing features of conditional random fields // Uncertainty in Artificial Intelligence, Proceedings of the Nineteenth Conference, UAI03, Acapulco, Mexico, August 7–10, 2003. Morgan Kaufmann, 2003. P. 403–410.

11. Feldman R., Sanger J. Probabilistic Models for Information Extraction // The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, 2006. P. 131–145.

12. Wang A., Singh A., Michael J., *et al.* GLUE: a multi-task benchmark and analysis platform for natural language understanding // Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, Louisiana, USA, May 6–9, 2019. P. 1–20. DOI: 10.18653/v1/w18-5446.

13. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, Minnesota, USA, June 2–7, 2019. Vol. 1: Long and Short Papers. Association for Computational Linguistics, 2019. P. 4171–4186. DOI: 10.18653/v1/n19-1423.

14. Pennington J., Socher R., Manning C.D. Glove: Global vectors for word representation // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Doha, Qatar, October 25–29, 2014. Association for Computational Linguistics, 2014. P. 1532–1543. DOI: 10.3115/v1/d14-1162.

15. Burtsev M., Seliverstov A., Airapetyan R., *et al.* DeepPavlov: Open-Source Library for Dialogue Systems // Association for Computational Linguistics-System Demonstrations, Proceedings of the 56th Annual Meeting, Melbourne, Australia, July 15–20, 2018. Association for Computational Linguistics, 2018. P. 122–127. DOI: 10.18653/v1/p18-4021.

16. Xue K., Zhou Y., Ma Z., *et al.* Fine-tuning BERT for joint entity and relation extraction in Chinese medical text // Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM, San Diego, California, USA, November 18–21, 2019. IEEE, 2019. P. 892–897. DOI: 10.1109/bibm47256.2019.8983370.

17. Wang Q., Yang L., Kanagal B., *et al.* Learning to Extract Attribute Value from Product via Question Answering: A Multi-task Approach // Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD'20, USA, August 23–27, 2020. Association for Computing Machinery, 2020. P. 47–55. DOI: 10.1145/3394486.3403047.

18. Banerjee P., Pal K.K., Devarakonda M.V., Baral C. Biomedical Named Entity Recognition via Knowledge Guidance and Question Answering // ACM Transactions on Computing for Healthcare. 2021. Vol. 2, no. 4. P. 1–24. DOI: 10.1145/3465221.

19. Li X., Yin F., Sun Z., *et al.* Entity-Relation Extraction as Multi-Turn Question Answering // Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 – August 2, 2019. Vol. 1: Long Papers. Association for Computational Linguistics, 2019. P. 1340–1350. DOI: 10.18653/v1/p19-1129.

20. Qiu L., Ru D., Long Q., *et al.* QA4IE: A Question Answering Based Framework for Information Extraction // Proceedings of the 17th International Semantic Web Conference, ISWC 2018, Monterey, California, USA, October 8–12, 2018. Vol. 11136 / ed. by D. Vrandečić, K. Bontcheva, M.C. Suárez-Figueroa, *et al.* Springer, 2018. P. 198–216. Lecture Notes in Computer Science. DOI: 10.1007/978-3-030-00671-6_12.

21. Rajpurkar P., Jia R., Liang P. Know What You Don't Know: Unanswerable Questions for SQuAD // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018. Vol. 2: Short Papers. Association for Computational Linguistics, 2018. P. 784–789. DOI: 10.18653/v1/p18-2124.

22. Weischedel R., Hovy E., Marcus R., *et al.* OntoNotes: A large training corpus for enhanced processing // Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation / ed. by J. Olive, C. Christianson, J. McCary. Springer, 2011.

23. Google Research Github Account. TensorFlow code and pre-trained models for BERT. URL: `https://github.com/google-research/bert` (дата обращения: 31.10.2022).

24. DeepPavlov lab Github Account. An open source library for deep learning end to end dialog systems and chatbots. URL: `https://github.com/deeppavlov/DeepPavlov` (дата обращения: 31.10.2022).

25. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Hong Kong, China, November 3–7, 2019. Association for Computational Linguistics, 2019. P. 3982–3992. DOI: 10.18653/v1/D19-1410.

26. Ubiquitous Knowledge Processing Lab Github Account. Multilingual Sentence & Image Embeddings with BERT. URL: `https://github.com/UKPLab/sentence-transformers` (дата обращения: 31.10.2022).

27. An open source machine learning framework PyTorch. URL: `https://pytorch.org/` (дата обращения: 31.10.2022).

Березкин Дмитрий Валерьевич, к.т.н., доцент, кафедра «Компьютерные системы и сети», Московский государственный технический университет имени Н.Э. Баумана (Москва, Российская Федерация)

Козлов Илья Андреевич, магистр, младший научный сотрудник, научно-учебный комплекс «Информатика и системы управления», Московский государственный технический университет имени Н.Э. Баумана (Москва, Российская Федерация)

Мартынюк Полина Антоновна, магистрант, кафедра «Компьютерные системы и сети», Московский государственный технический университет имени Н.Э. Баумана (Москва, Российская Федерация)

Панфилкин Артем Михайлович, магистрант, кафедра «Компьютерные системы и сети», Московский государственный технический университет имени Н.Э. Баумана (Москва, Российская Федерация)