Original article
DOI: 10.14529/ctcr220203

METHODOLOGY FOR SOLVING PROBLEMS OF CLASSIFICATION OF APPEALS/REQUESTS OF CITIZENS TO THE "HOTLINE" OF THE PRESIDENT OF THE RUSSIAN FEDERATION

***E.V. Bunova**, bunovaev@susu.ru, https://orcid.org/0000-0002-0997-8000*
***V.S. Serova**, vladislava.serova.98@gmail.com, https://orcid.org/0000-0001-9045-1048*
South Ural State University, Chelyabinsk, Russia

Abstract. The use of neural networks for the classification of text data is an important area of digital transformation of socio-economic systems. The article is devoted to the description of the methodology for classifying citizens' appeals. The proposed technique involves the use of a convolutional neural network. The stages of processing citizens' appeals in the amount of 7000 appeals are described. In order to reduce the dimension of the problem, methods of filtering and removing stop words were applied. The resulting data set allows you to choose the best classifier in terms of accuracy, specificity, sensitivity. Training and test samples were used, as well as cross-validation. The article shows the effectiveness of using this method to distribute requests on 15 topics of citizens' appeals to the "hotline" of the President of the Russian Federation. Automating the classification of received appeals by topic allows them to be processed quickly for further study by the relevant departments. **The purpose of the study** is automation of the distribution of citizens' appeals to the President's hotline by category based on the use of modern machine learning methods. **Materials and methods.** The development of software that automates the process of distributing citizens into categories is carried out using a convolutional neural network written in the Python programming language. **Results**. With the help of the prepared data set, the pre-trained model of NL BERT and sciBERT was trained by the deep learning method. The model shows an accuracy of 86% in the estimates of quality metrics. **Conclusion**. A pre-trained model was trained using a convolutional neural model using a prepared data set. Even if the forecast does not match the real category, the model gives a minor error, correctly determines the category of the appeal. The results obtained can be recommended for practical application by authors of scientific publications, scientific institutions, editors and reviewers of publishing houses.

Keywords: text processing, machine learning, convolutional neural networks, categorization of text, LSTM, CNN, deep learning, text analysis

For citation: Bunova E.V., Serova V.S. Methodology for solving problems of classification of appeals/requests of citizens to the "hotline" of the President of the Russian Federation. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*. 2022;22(2):29–40. DOI: 10.14529/ctcr220203

Научная статья
УДК 004.85
DOI: 10.14529/ctcr220203

МЕТОДИКА РЕШЕНИЯ ЗАДАЧ КЛАССИФИКАЦИИ ОБРАЩЕНИЙ/ЗАПРОСОВ ГРАЖДАН НА «ГОРЯЧУЮ ЛИНИЮ» ПРЕЗИДЕНТА РФ

***Е.В. Бунова**, bunovaev@susu.ru, https://orcid.org/0000-0002-0997-8000*
***В.С. Серова**, vladislava.serova.98@gmail.com, https://orcid.org/0000-0001-9045-1048*
Южно-Уральский государственный университет, Челябинск, Россия

Аннотация. Применение нейронных сетей для классификации текстовых данных является важной сферой цифровой трансформации социально-экономических систем. Статья посвящена описанию методики классификации обращений граждан. Предлагаемая методика включает использование сверточной нейронной сети. Описаны этапы обработки обращений граждан в количестве 7000 обращений.

С целью сокращения размерности задачи применены методы фильтрации, удаления стоп-слов. Полученный набор данных позволяет выбрать лучший классификатор по показателям точности, специфичности, чувствительности. Использованы обучающая и тестовая выборки, а также кросс-валидация. В статье показана эффективность использования данного метода для распределения запросов по 15 темам обращений граждан на «горячую линию» Президента РФ. Автоматизация классификации поступивших обращений по темам позволяет быстро их обработать для дальнейшей проработки соответствующих ведомств. **Целью исследования** является автоматизация распределения обращений граждан на горячую линию Президента по категориям на основе использования современных методов машинного обучения. **Материалы и методы.** Разработка программного обеспечения, автоматизирующего процесс распределения граждан по категориям, осуществляется с использованием сверточных нейронных сетей, написанных на языке программирования Python. **Результаты.** С помощью подготовленного набора данных предварительно обученная модель NL BERT и sciBERT была обучена методом глубокого обучения. Модель показывает точность 86 % в оценках показателей качества. **Заключение.** В ходе исследования с помощью подготовленного набора данных была обучена методом использования сверточной нейронной предобученная модель. Даже при несовпадении прогноза с реальной категорией модель дает незначительную ошибку, правильно определяет категорию обращения. Полученные результаты могут быть рекомендованы для практического применения авторами научных публикаций, научными учреждениями, редакторами и рецензентами издательств.

Ключевые слова: обработка текста, машинное обучение, сверточные нейронные сети, категоризация текста, LSTM, CNN, глубокое обучение, анализ текста

Для цитирования: Bunova E.V., Serova V.S. Methodology for solving problems of classification of appeals/requests of citizens to the “hotline” of the President of the Russian Federation // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2022. Т. 22, № 2. С. 29–40. DOI: 10.14529/ctcr220203

Introduction

Currently, a lot of attention from the federal authorities and the leadership of the Russian Federation is paid to improving the “quality of life” of the population. In his Messages to the Federal Assembly, Russian President Vladimir Putin has repeatedly stated the need to improve the standard of living, ensure a decent, long life for Russians and improve its quality as the goal of the socio-economic development of the country and the implementation of National projects [1, 2]. The targets for improving the quality of life of Russians were formulated in Presidential Decree No. 204 of May 7, 2018 “On National goals and strategic objectives for the development of the Russian Federation for the period up to 2024” [3]. On July 13, 2020, at a meeting of the Council for Strategic Development and National Projects, Russian President Vladimir Putin said: “… we will discuss the key directions of the country's development, our further actions, and their main, unifying task is to improve the quality of life of citizens. I want to emphasize once again: a person should be at the center of all our decisions, plans, and programs” [4].

An important event held by the President of Russia V.V. Putin is the annual “hotline”, to which every citizen can send an appeal/request for solving urgent problems for him. So, for example, only in the Chelyabinsk region for a few days, about 4 thousand requests from citizens are received by the “hotline” of the President of the Russian Federation.

It is impossible to process these requests quickly using manual methods, and some appeals/requests from citizens require a quick response. Therefore, an urgent task is to develop a software application to automate the classification of incoming requests by topic and send these requests to the relevant structures that are authorized to solve the problems described by citizens.

This article describes the solution to the problem of classifying appeals /requests of citizens by topics, namely: COVID, Highways, Alimony, Banks and loans; Landscaping; Veterinary medicine; Water supply, Issues of remuneration and employment, Issues of pensions and retirement experience, Gas supply, Citizenship, Courtyards and common areas, Housing, Healthcare of the Russian Federation, Healthcare and medical care in the Chelyabinsk region.

These topics are the most popular, the number of requests/queries on these topics is about 70% of the total number of requests.

Currently, the following methods are most often used to classify text data by topic:

1. Define the document topic manually. The method is accurate, but usually has such a disadvantage as the inability to process large volumes in a sufficient amount of time, and there is also subjectivity in data processing. Manual classification is very limited in the ability to quickly process large arrays of texts, characteristic of many applications of automatic text classification methods. Such methods are widely used by modern Internet systems: news aggregators, such as the Yandex service.News or Google News to solve the problem of thematic classification of documents and news stories, email services, for example, Yandex.Mail, gmail, or Mail.ru use algorithms detect spam filter mail, search engines (Yandex, Google, Mail.ru, Yahoo and others) resolve the challenge of ensuring diversity of search results, etc. [5].

2. Determining the topic of the document automatically using the developed rules based on regular expressions. The method allows you to process large amounts of data, but requires efforts to develop and maintain the rules up to date. In addition, before defining the rules, a specialist is obliged to familiarize himself in depth with various data samples of all topics on which a large amount of time can be spent. The disadvantage of the described method is its high sensitivity to errors that may occur accidentally or systematically both during the digitization of the text and during its formation. It is the person who is the main factor of non-determinism when placing bibliographic information in texts [6].

3. Determining the topic of the document automatically using machine learning. When using this approach, the dependency of the theme on the sample is set automatically. Manual markup of the training sample is required beforehand, but this is often a simpler task than finding the rules for belonging to the topics of all samples. This approach is currently the most promising.

The direction of using machine learning is currently widely used. For example, machine learning is used in law enforcement agencies when employees receive tactical recommendations [7]. Artificial intelligence is already being introduced into medical institutions. For example, the processing of patient data, preliminary diagnosis and even the selection of individual treatment is implemented on the basis of information about a person's illness.

Implemented machine learning algorithms make predictions or make decisions not on the basis of strictly static program commands, but on the basis of a training sample, with the help of which the parameters of the model are adjusted. Various branches of mathematics are used for the process of setting up (fitting) a model based on data sampling: mathematical statistics, optimization methods, numerical methods, probability theory, linear algebra, mathematical analysis, discrete mathematics, graph theory, various techniques for working with digital data, etc. The result of the learning algorithm is a function that approximates (restores) an unknown dependence in the processed data.

When talking about machine learning, they often mean artificial neural networks (hereinafter – INS) and deep learning, which have become popular, which are machine learning models presented in Fig. 1, i.e. special cases of pattern recognition methods, discriminant analysis, clustering methods, etc.

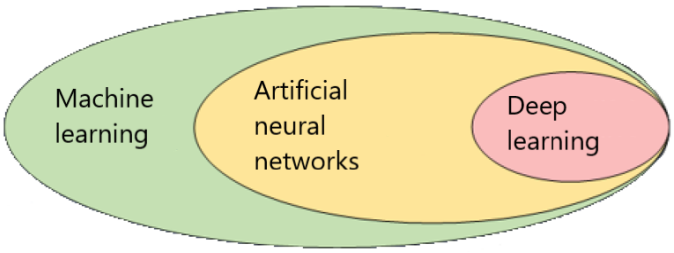

Fig. 1. Machine learning models

One of the machine learning models is INS. Currently, there is a revival of INS under the new brand "Deep Learning" (Deep Learning). So, in the article [8], with the help of deep learning, stroke risk factors are extracted from medical texts. Based on the results of the experiments, conclusions were drawn about the effectiveness of the developed methods and the text characteristics used to solve the problem.

A similar approach was used in the article [9] for clustering the corpus of documents in Russian. The results of applying the algorithms are demonstrated in the work on real data and show the high accuracy of the chosen method.

INS are hierarchical classifiers that are able to independently identify features in the original signal. A common indicator of the INS is the number of hidden layers. Some modern networks have hundreds or even thousands of hidden layers. There are a large number of INS architectures. Let's list the most popular of them.

Networks without feedbacks or direct signal propagation networks in which the signal passes from the outputs of the neurons of the i-th layer to the inputs of the neurons of the $(i+1)$-th layer and does not return to the previous layers:

– perceptrons (single-layer, multi-layer with cross-links, etc.), except perceptrons with feedbacks;

– Bayesian neural network;

– extreme learning machine;

– in fact, any INS that is a directed acyclic (without cycles) graph.

The article [10] reflects the disadvantages and advantages of these networks. The advantages of networks without feedbacks are the simplicity of their implementation and guaranteed receipt of a response after passing data through layers.

The disadvantage of this type of network is the minimization of the size of the network – neurons repeatedly participate in data processing.

Convolutional Neural Networks (SNN, Connews), a distinctive feature of which is the convolution operation:

– AlexNet;

– LeNet-5;

– convolutional networks with region allocation (Region Based CNNs, R-CNN);

– deploying neural networks (deconvolutional networks, DN, DeConvNet) or reverse graphic networks, convolutional networks on the contrary.

In the article [11], the process of determining the subject of texts is automated using a convolutional neural network of deep learning.

The methods and tools used in the construction of a neural network for semantic classification of text are described in the article by authors V.I. Voronov and E.V. Martynenko [12].

The authors Y.V. Kotenko, S.A. Petrenko [13] described an approach to assessing the reliability of information posted on a social network. The reliability of information is considered from the point of view of its truth. It is proposed to evaluate the reliability of the information provided in the social network entry using classification algorithms. It is proposed to use convolutional neural networks to analyze the texts of records. The article also describes an algorithm for constructing and using a tool for assessing reliability, as well as possible options for its application.

Generative adversarial networks (hereinafter referred to as GAN), which consist of two competing INS: a generative model that generates samples, and a discriminative model that tries to distinguish correct (“genuine”) samples from incorrect ones. GAN is quite difficult to train, because the task is not just to train two networks, but also to maintain a balance, an equilibrium between them. If one of the networks (generator or discriminator) becomes much better than the other, then the GAN will not converge (learn).

The author U.D. Muratova [14] considered the development tools necessary for the implementation of an information system based on the analysis of text perception.

The disadvantage of GAN is the long process of learning the model [15].

Recurrent Neural Networks (SNN) or networks with memory. They contain neurons that can store information about their previous states during operation, such neurons receive information not only from the previous layer, but also from themselves as a result of the previous passage. Recurrent networks are the neural network embodiment of Markov chains. There are many architectures of recurrent INS:

– network with long-term and short-term memory (Long Short Term Memory, LSTM);

– fully recurrent network;

– recursive network;

– Hopfield neural network, a type of fully connected INS;

– Boltzmann machine and limited Boltzmann machine;

– Hamming neural network;

– Bidirectional associative memory (BAM) or Kosko neural network;
– bidirectional recurrent neural networks (bidirectional recurrent neural networks);
– Elman and Jordan networks;
– echo-networks and impulse (spike) neural networks;
– unstable state machines (liquid state machines, LSM);
– neural history compressor;
– recurrent networks of the second order;
– controlled recurrent neurons (Gated Recurrent Units, GRU);
– neural Turing machines (Neural Turing machines, NTM), etc.

In this article [16], recurrent neural networks are used in natural language text processing tasks.

The main disadvantage of these networks [17] is the lack of stability, and in cases when it is achieved, the network becomes equivalent to a single-layer neural network, which is why it is unable to solve linearly inseparable problems. As a result, the capacity of such networks is extremely small.

Convolutional neural networks have proven themselves well in the tasks of object recognition and machine vision. This has led to further research into the way they are applied, one of which is the task of classifying the text.

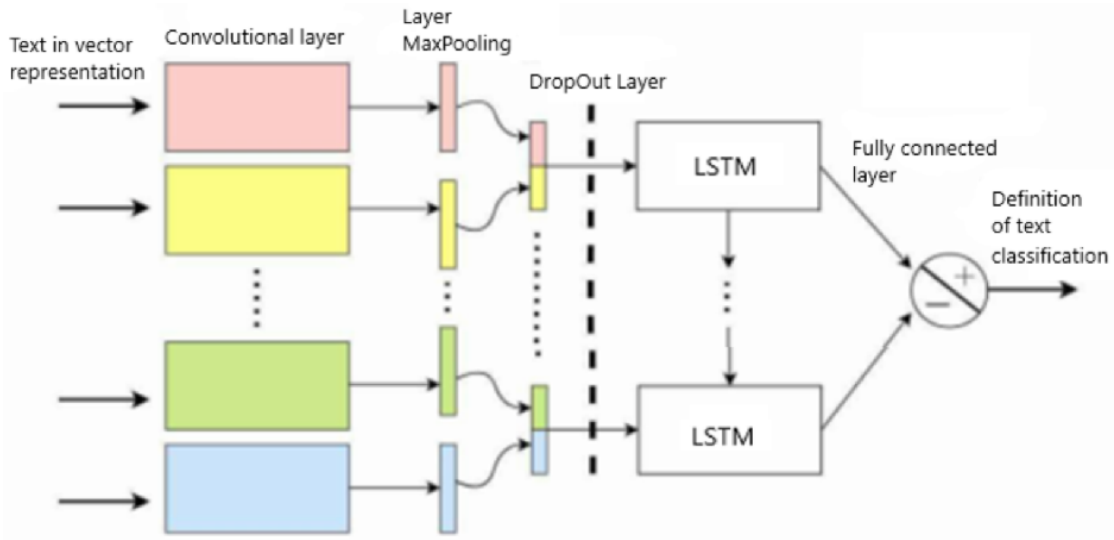

Fig. 2. Architecture of the designed neural network

To understand the architecture of the designed neural network, shown in Fig. 2, let's consider each layer separately:

– A convolutional layer is a layer that consists of a feature map, each map has its own core. The number of feature maps is determined by the requirements for the task, if you take a large number of maps, the accuracy of the model will increase, but the computational complexity will increase. The core is a filter or window that "slides" over the entire area of the previous map and finds certain signs.

– A subdiscretizing layer is a layer that performs a dimensionality reduction of the input feature map. The convolution layer has already identified some features, then for subsequent processing such a detailed feature map is no longer needed, and it is compacted to a less detailed one. In addition, filtering out unnecessary details helps to avoid retraining.

– A fully connected layer is a layer in which each neuron is connected to all the neurons on the previous layer, and each connection has its own weighting factor. In the Keras library, this layer has the name "Dense".

– Dropout layer is a way to combat retraining in neural networks. This layer excludes a certain percentage (for example, 20%) of random neurons (located in both hidden and visible layers) at different iterations during neural network training. This technique significantly increases the learning rate, the quality of training on training data, and also improves the quality of model predictions on new test

data. In the architecture being developed, which is based on a convolutional neural network, cores of different sizes will be used, which are designed for parallel processing of the n-gram of text, respectively. After processing by convolution layers, feature maps arrive at the subdiscretization layers, which extract the most significant n-grams from the text. After that, it is combined into a common feature vector. Then the resulting vector is fed into a hidden fully connected layer. At the last step, the resulting feature map is fed to the output layer of a neural network with a sigmoidal activation function. The number of consecutive convolutional layers, the size of the cores of the convolutional layer and the subdiscretization is determined experimentally. Kernels of sizes 1, 2, 3, 4 and 5 are designed to process one word, bigrams, trigrams, 4-grams and 5-grams, respectively.

Let's consider a methodology for solving problems of classifying citizens' appeals/requests by topic, developed on the basis of a convolutional neural network.

The methodology for classifying citizens' appeals/requests by topic **consists of the following stages**:

Stage 1. Preliminary preparation of the data set.

When developing the classification model, data on appeals/requests of citizens to the “hotline” of the President of the Russian Federation living in the Chelyabinsk region were used. The number of requests/requests is about 7 thousand records.

Preparation of a set of this includes:

1. Clearing text data from unnecessary characters.
2. Converting text to lowercase.
3. Perform tokenization, normalization and filtering of text data.

Tokenization involves dividing the text into words in accordance with regular expressions, the specified template for which allows you to remove punctuation marks from the text. We will set the maximum number of tokens to be taken into account during processing, as well as the relative frequencies of token use that occur in the analyzed text. This allows you to exclude rare, as well as very frequently used words that will lead to the exclusion or retraining of the program. Solving the problem of normalization (lemmatization) allows you to bring the words selected as a result of tokenization to a normal form. Only single terminals of the analyzed text will participate in further processing.

To filter the data, a dictionary of words was created, shown in Fig. 3, which do not affect the definition of the category of treatment and were automatically removed without loss of semantic content from further processing of text data. The size of this dictionary is 28% of the total amount of text data.

```
replace_vocab={"аннотация" : " ",# словарик слов, которые не несут нагрузки
"<person>ович" : " ", "<person>вна" : " ",
"здравствуйте" : "", "добрый день" : " ", "здравия желаю" : " ", "добрый вечер": " ", "Доброго времени суток": " ", "Здравств
"уважаемый" : " ", "ув " : " ", " с ув " : " ", " в в " : " ", "Уважаемый Президент РФ" : " ",
"<person><person>" : "<person>", "<person><person><person>" : "<person>",
"Сёмкина" : " ", "Азаркин" : " ", "ТУРБАЛ" : " ", "СЕМЬЯ НЕСВИТ" : " ", "Землянская" : " ", "Чикалова" : " ", "Л В": " ",
"Кустов" : " ", "Клён" : " ", "Будяк  Надежда" : " ", "Монахов" : " ", "София" : " ", "Романов" : " ", "Вера" : " ",
"Любовь" : " ", "Валов" : " ", "Тишунова" : " ", "Твердохлеб" : " ", "Слущащ" : " ", "Серго" : " ", "Харюшина" : " ",
"Мажитова" : " ", "Лодвикова" : " ", "Азаркин" : " ", "Логовчина" : " ", "Чикалова" : " ", "Монахов" : " ", "Трусов" : " ",
"Лопин" : " ", "Червяковой Т  Н" : " ", "Червяковой О  Н" : " ", "Щегликов" : " ", "Щегликова В М" : " ", "Л П": " ", "Шумски
"Галайда": " ", "Сагитова": " ", "В В Путен": " ", "Торбин" : " ", "Мегрибан Гачай кызы" : " ", "Машина Надежда" : " ",
"врач Марьина <person> <person>" : " ", "Злоказова" : " ", "Фанизовна" : " ",
"Почта  tveranastasia gmail com" : " ",
"зовут меня" : " ", "меня зовут" : " ", "представлюсь" : " ",
"года рождения" : " ", "дата рождения" : " ",
"Вызов №  тел    моб  тел   " : " ",
"Очень хочется  чтобы вы прочитали мое обращение" : " ", "Очень надеюсь" : " ", "Скажите  пожалуйста" : " ", "Спасибо" : " "
"Заранее спасибо" : " ", "Дело в том  что" : " ", "просьба" : " ", "Прошу помочь в <person> проблеме" : " ",
"Разве это справедливо  <person> <person>ович  вы же Президент  Вы можете решить нашу проблему" : " ", "Это же абсурд" : "
"Очень прошу Вас решить эту проблему" : " ", "Заранее большое спасибо" : " ", "Спасибо за внимание" : " ", "прошу" : " ",
"Прошу разобраться" : " ", "обращаюсь к вам с просьбой о  помощи" : " ", "просьба" : " ", "Обратите внимание" : " ", "благода
"Я понимаю что у Вас много дел и Вы вряд ли будете заниматься этим сами  но может поручите кому нибудь разобраться с нашей пр
```

Fig. 3. A fragment of a dictionary of words that do not carry semantic words

Stage 2. Model training.

To conduct the training of the model, a dictionary of keywords was created to distribute citizens' appeals into 15 categories, which are the most in demand. The number of requests/queries on these topics is about 70% of the total number of requests.

A fragment of the dictionary is shown in Fig. 4.

```
In [234]: def text_update_key(s):
              vocab=['дороги', 'трамваи', 'рельсы', 'асфальт', 'тросы', 'безопасность', 'пешеходы', 'мост', 'освещение', 'отсыпка', 'грейде
                     'светофор', 'придорожный', 'сервис', 'шум', 'реагенты', 'парковка', 'подъездной' 'путь', 'яма', 'общественный', 'транс
                     'выхлопы', 'многодетная', 'семья', 'инвалид', 'ветеран', 'волонтеры', 'алименты', 'алиментщик', 'родительские',
                     'права', 'судебные', 'приставы', 'вкладчики', 'ипотека', 'мошенничество', 'накопления', 'каникулы', 'кредит',
                     'налог', 'вклад', 'девальвация', 'Сбербанк', 'обязательства', 'списание', 'пенсионные', 'взносы', 'комиссия',
                     'рефинансирование', 'заработная', 'плата', 'пирамида', 'долг', 'проценты', 'счет', 'сберегательная', 'книжка', 'банк',
                     'компенсация', 'потребительский', 'кооператив', 'карта', 'коллекторы', 'банкрот', 'наличные', 'индексация',
                     'благодарность', 'глава', 'города', 'губернатор', 'благоустройство', 'околошкольная', 'территория', 'детская', 'площад
                     'преображение', 'снег', 'мусор', 'деревья', 'памятник', 'чистить', 'поиск', 'работы', 'интернет',
                     'собак', 'пляж', 'городской', 'парк', 'аттракционы', 'радиочастотная', 'электромагнитная', 'антенна', 'зловоние', 'пит
                     'вода', 'дворец', 'спорта', 'детский', 'садик', 'поликлиники', 'придомовая', 'территория', 'облагораживание', 'очистны
                     'сооружения', 'очистка', 'реки', 'канализации', 'стоянки', 'автомобилей', 'тротуар', 'межевание', 'двора', 'незаконные
                     'лесной', 'массив', 'приют', 'выгул', 'пчелы', 'вода', 'трубы', 'водопровод', 'ЖКХ', 'водоснабжение',
                     'водоотведение', 'трудоустройство', 'инвалиды', 'работодатели', 'работа', 'индексирование', 'инфляция',
                     'пенсия', 'инвалидность', 'оплата', 'МРОТ', 'пособия', 'сокращение', 'РВП', 'вахтовый', 'метод', 'центр',
                     'занятости', 'бюджетники', 'неофициально', 'сокращение', 'социальная', 'польза', 'оклад', 'прожиточный', 'минимум',
                     'трудовой', 'стаж', 'должность', 'изобретательство', 'производство', 'цены', 'средний', 'доход',
```

Fig. 4. A fragment of the keyword dictionary

Next, we will apply the apply() function of reducing all words to lowercase, the text_update_key() function to the entire array of text data and the onlygoodsymbols() function using the apply function, which is used in cases when it is necessary to apply any function to all rows or columns of the matrix (or arrays of larger dimension). The code of these functions is shown in Fig. 5.

```
X= X.apply(get_lower)
X= X.apply(text_update_key)
X= X.apply(onlygoodsymbols)
```

Fig. 5. Application of the apply function

Fig. 6 shows the result of this function.

```
: print(X.head())

0    лет города асфальт суд суд
6                        ремонт
7             лет лет газ жилье
8                           лет
9                    лес детский
Name: Текст обращения, dtype: object
```

Fig. 6. A fragment of the processed data

We use the train_test_split module, shown in Fig. 7, of the Scikit-learn library, which is useful for separating datasets, and to avoid problems with retraining, we divide the data set:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(
     X, y, test_size=0.1, stratify=y, random_state=42)
```

Fig. 7. Using the train_test_split module

Fig. 8 represents the output of the first line from x_train.

```
In [244]: x_train[0]

Out[244]: 'лет города асфальт суд суд'
```

Fig. 8. Output of the first line from x_train

To create a Sequential model, we import the libraries of optimizers Adam, RMSProp, SGD. First of all, the optimizer is a method of achieving the best results, helping to accelerate learning. In other words, it is an algorithm used to slightly change parameters such as weights and learning rate so that the model

works correctly and quickly. It uses a first-order moment estimation and a second-order gradient moment estimation to dynamically adjust the learning rate of each parameter. The main advantage of Adam is that after correcting the bias, each iteration of the learning rate has a certain range, which makes the parameters relatively stable, also among the advantages of the optimizer can be distinguished: simple implementation, computational efficiency and small memory requirements. The RMSProp algorithm calculates only the corresponding average value, so this can alleviate the problem of the algorithm's rapid learning rate decrease. The stochastic Gradient descent (SGD) algorithm reads part of the data and immediately calculates the gradient of the cost function to update the parameters.

We also import the callbacks class, shown in Fig. 9. Callback is a set of functions used at certain points during the training procedure. Callback functions are used to get information about the internal state of the model during training. You need to pass a list of callbacks (named with the callbacks argument) to the method.fit() Sequential or Model classes. Suitable callback methods will be called at each stage of training.

```
from tensorflow.keras.optimizers import Adam, RMSprop, SGD
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
```

Fig. 9. Importing optimizers and a class of callbacks.

Next, a Sequential model was created, which is a linear stack of layers that we will add using the .add() method, where Dense(1024), Dense(512), Dense(32) is a fully connected layer with 1024, 512 and 32 hidden neurons, respectively. Theoretically, the number of hidden layers can be arbitrarily large. Then we specify the training configuration (optimizer, loss function, metrics). It is necessary to choose the optimal size of the number of training facilities (batcha). The model is trained in this way: split the data into “packets” of batch_size size and sequentially iterate the entire dataset with a given number of “epochs”. It should be taken into account that with large batch_size sizes, there may not be enough memory on the video card, with too small sizes, training will be unstable.

The creation of a model and layers for it, as well as training with the optimizer RMSProp is shown in Fig. 10.

```
model = Sequential()


model.add(Dense(1024,activation='relu'))
model.add(Dropout(.3))
model.add()
model.add(Dense(512,activation='relu'))
model.add(Dropout(.3))
model.add(Dense(32,activation='relu'))
model.add(Dropout(.3))
model.add(Dense(y.shape[1], activation='softmax'))

model.compile(optimizer=RMSprop(momentum=.9,learning_rate=.0001), loss='categorical_crossentropy', metrics=['acc'])
# print(model.summary())
history = model.fit(x_train, y_train, epochs=200, batch_size=512, validation_split=0.2,callbacks = [early_stop, reduce_lr],verbos
```

Fig. 10. Creating and training a model

The process of learning the model, shown in Fig. 11, is taking place.

```
Epoch 1/200
6/6 [==============================] - 1s 51ms/step - loss: 2.0754 - acc: 0.1394 - val_loss: 2.0387 - val_acc: 0.2079 - lr: 1.0
000e-04
Epoch 2/200
6/6 [==============================] - 0s 14ms/step - loss: 2.0207 - acc: 0.2047 - val_loss: 1.9600 - val_acc: 0.2944 - lr: 1.0
000e-04
Epoch 3/200
6/6 [==============================] - 0s 16ms/step - loss: 1.9340 - acc: 0.2905 - val_loss: 1.8378 - val_acc: 0.3566 - lr: 1.0
000e-04
```

Fig. 11. Model learning process

Fig. 12 shows the accuracy metrics of the model training.

```
# print(np.argmax(model.predict(x_test),axis=-1),np.argmax(y_test,axis=-1))
print(model.evaluate(x_test,y_test))
```

```
12/12 [==============================] - 0s 4ms/step - loss: 1.0722 - acc: 0.8612
[1.0721559524536133, 0.8612021923065186]
```

Fig. 12.Model learning accuracy

In deep learning, loss is a value that a neural network tries to minimize: this is the distance between the true value and the predictions. To minimize this distance, the neural network learns by adjusting weights and offsets in such a way as to reduce losses, and the acc shows the percentage of instances that are correctly classified.

Thus, the evaluation of the quality of the model on the test sample is 86%.

The result of training the model. With a batch size of 512, the model under study needed 200 iterations (batches) for one training epoch.

Testing a trained model. Let's demonstrate how the model works on test data. To do this, we will create a separate csv file, which will contain 20% of the entire sample. The code for reading the file path is shown in Fig. 13.

```
obrashenie=pd.read_excel('/content/Test.xlsx')
obrashenie=obrashenie['Обращение']
```

Fig. 13. Reading the path to the test file

Let's output a list of requests. This list is shown in Fig. 14.

```
obr
```

```
0       ннотация  дравствуйте  формил кредит в овкомба...
1       ннотация  отрудники полиции  женщины  не могут...
2       ннотация  очему в детском саду у воспитателей ...
3       ннотация  ромадные тарифы на холодное водоснаб...
4       ннотация  плата труда медицинских работников  ...
                                ...
1448     с моей семьей проживаю в г  оркино елябинской...
Name: Текст обращения, Length: 1453, dtype: object
```

Fig. 14. Output of requests in the test file

The result of training the model is shown in Fig. 15.

```
print(np.argmax(pred,axis=-1))
for i in np.argmax(pred,axis=-1):
  print(ly[i])
```

```
[4 5 1 ... 4 4 4]
Льготы и соц. помощь
Многоквартирные дома
Вопросы пенсий и пенсионного стажа
Многоквартирные дома
Образование
COVID
Льготы и соц. помощь
Вопросы пенсий и пенсионного стажа
Жилье
Жилье
Природа, Экология
Жилье
Образование
Природа, Экология
Вопросы пенсий и пенсионного стажа
Льготы и соц. помощь
```

Fig. 15.The result of model training

Conclusions

With the help of a prepared data set, a pre-trained model of NL BERT and sciBERT was trained by the deep learning method. The model shows an accuracy of 86% in the estimates of quality metrics.

The results obtained can be recommended for practical application by authors of scientific publications, scientific institutions, editors and reviewers of publishing houses.

References

1. *Poslaniye Prezidenta Federal'nomu Sobraniyu 15 yanvarya 2020 goda* [The President's Message to the Federal Assembly on January 15, 2020]. Available at: http://www.kremlin.ru/events/president/news/62582 (accessed 20.12.2021). (In Russ.)

2. *Poslaniye Prezidenta Federal'nomu Sobraniyu 20 fevralya 2019 goda* [The President's Message to the Federal Assembly on February 20, 2019]. Available at: http://www.kremlin.ru/events/president/news/59863 (accessed 20.12.2021). (In Russ.)

3. *Ukaz Prezidenta Rossiyskoy Federatsii ot 07.05.2018 g. N 204 "O natsional'nykh tselyakh i strategicheskikh zadachakh razvitiya Rossiyskoy Federatsii na period do 2024 goda". Vstupil v silu s 7 maya 2018 goda* [Decree of the President of the Russian Federation No. 204 dated 07.05.2018 "On national goals and strategic objectives of the development of the Russian Federation for the period up to 2024". Entered into force on May 7, 2018]. Available at: http://www.kremlin.ru/acts/bank/43027 (accessed 20.12.2021). (In Russ.)

4. *Zasedaniye Soveta po strategicheskomu razvitiyu i natsional'nym proyektam 13 iyulya 2020 goda* [Meeting of the Council for Strategic Development and National Projects on July 13, 2020]. Available at: http://www.kremlin.ru/events/president/news/63635 (accessed 20.12.2021). (In Russ.)

5. Shagraev A.G. *Modifikatsiya, razrabotka i realizatsiya metodov klassifikatsii novostnykh tekstov: avtoref. dis. kand. tekhn. nauk* [Modification, development and implementation of methods of classification of news texts. Abstract of Cand. diss.]. Moscow; 2014. 19 p. (In Russ.)

6. Sokolova T.A. An extraction of the elements from bibliography based on automatically generated regular expressions. *Information and telecommunication technologies and mathematical modeling of high-tech systems: Materials of the All-Russian conference with international participation*. Moscow; 2019. P. 313–316. (In Russ.)

7. Ushakov O.V. [Application of automated information systems with machine learning integration in law enforcement agencies]. *Problemy pravovoy i tekhnicheskoy zashchity informatsii*. 2018;(6):142–147. (In Russ.)

8. Donitova V.V., Kireev D.A., Titova E.V., Akimova A.A. Natural language processing models for extraction of stroke risk factors from electronic health records. *Trudy Instituta sistemnogo analiza Rossiyskoy akademii nauk = Proceedings of the Institute of system analysis of the Russian academy of sciences*. 2021;71(4):93–101. (In Russ.) DOI: 10.14357/20790279210410

9. Kolmogortsev S.V., Sarayev P.V. [Bibliography extraction from texts by regular expressions]. *Novyye informatsionnyye tekhnologii v avtomatizirovannykh sistemakh*. 2017;(20):82–88. (In Russ.)

10. Gorbachevskaya E.N. Classification of neural networks. *Vestnik Volzhskogo universiteta im. V.N. Tatishcheva*. 2012;2(19):128–134. (In Russ.)

11. Katenko Yu.V. Application of machine learning methods for text information analysis. *Okhrana, bezopasnost', svyaz'*. 2019;3(4):90–94. (In Russ.)

12. Voronov V., Martinenko E. Research of parallel structures of neural networks for use in the tasks on the Russian text semantic classification considering limited computing resources (on the example of operational reports used in the RF MIA). *Economics and Quality of Communication Systems*. 2018;3(9):52–60. (In Russ.)

13. Katenko Yu.V., Petrenko S.A. [The concept of control of the reliability of information in the professional social network using convolutional neural networks]. In: *Mezhdunarodnaya konferentsiya po myagkim vychisleniyam i izmereniyam. Vol. 1*. St. Petersburg; 2019. P. 140–143. (In Russ.)

14. Muratova U.D. [Studying neural networks for chatbots]. In: *Proceedings of the IX Congress of Young Scientists*. St. Petersburg; 2021. P. 92–95. (In Russ.)

15. Sukhan' A.A. Applying generative adversarial network to the problem of trend determenition. *Moskovskiy ekonomicheskiy zhurnal*. 2019;(6):180–191. (In Russ.) DOI: 10.24411/2413-046X-2019-16031

16. Budyl'skiy D.V. [Application of recurrent neural networks in processing natural language texts]. Voprosy nauki. 2015;6:8–12. (In Russ.)

17. Danchenko V.V. Overview of funds development of an information system based on analysis of text perception. *Informatika i prikladnaya matematika*. 2020;(26):31–34. (In Russ.)

Список литературы

1. Послание Президента Федеральному Собранию 15 января 2020 года [Электронный ресурс]. URL: http://www.kremlin.ru/events/president/news/62582 (дата обращения: 20.12.2021).

2. Послание Президента Федеральному Собранию 20 февраля 2019 года [Электронный ресурс]. URL: http://www.kremlin.ru/events/president/news/59863 (дата обращения: 20.12.2021).

3. Указ Президента Российской Федерации от 07.05.2018 г. № 204 «О национальных целях и стратегических задачах развития Российской Федерации на период до 2024 года». Вступил в силу с 7 мая 2018 года [Электронный ресурс]. URL: http://www.kremlin.ru/acts/bank/43027 (дата обращения: 20.12.2021).

4. Заседание Совета по стратегическому развитию и национальным проектам 13 июля 2020 года [Электронный ресурс]. URL: http://www.kremlin.ru/events/president/news/63635 (дата обращения: 20.12.2021).

5. Шаграев А.Г. Модификация, разработка и реализация методов классификации новостных текстов: автореф. дис. … канд. техн. наук. М., 2014. 19 с.

6. Соколова Т.А. Извлечение элементов библиографии на основе автоматически порождаемых регулярных выражений // Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем: материалы Всерос. конф. с междунар. участием. М., 2019. С. 313–316.

7. Ушаков О.В. Применение автоматизированных информационных систем с интеграцией машинного обучения в деятельности правоохранительных органов // Проблемы правовой и технической защиты информации. 2018. № 6. С. 142–147.

8. Методы обработки естественного языка для извлечения факторов риска инсульта из медицинских текстов / В.В. Донитова, Д.А. Киреев, Е.В. Титова, А.А. Акимова // Труды Института системного анализа Российской академии наук. 2021. Т. 71, № 4. С. 93–101. DOI: 10.14357/20790279210410

9. Колмогорцев С.В., Сараев П.В. Извлечение библиографии из текстов регулярными выражениями // Новые информационные технологии в автоматизированных системах. 2017. № 20. С. 82–88.

10. Горбачевская Е.Н. Классификация нейронных сетей // Вестник Волжского университета им. В.Н. Татищева. 2012. № 2 (19). С. 128–134.

11. Катенко Ю.В. Применение методов машинного обучения для анализа текстовой информации // Охрана, безопасность, связь. 2019. Т. 3, № 4 (4). С. 90–94.

12. Воронов В.И., Мартыненко Э.В. Исследование параллельных структур нейронных сетей для использования в задачах по семантической классификации текста на русском языке в условиях ограничения вычислительных ресурсов (на примере оперативных сводок в системе МВД России) // Экономика и качество систем связи. 2018. № 3 (9). С. 52–60.

13. Катенко Ю.В., Петренко С.А. Концепция контроля достоверности информации в профессиональной социальной сети с применением сверточной нейронной сети // Международная конференция по мягким вычислениям и измерениям. СПб., 2019. Т. 1. С. 140–143.

14. Муратова У.Д. Изучение нейронных сетей для чат-ботов // Материалы IX Конгресса молодых ученых. СПб., 2021. С. 92–95.

15. Сухань А.А. Генеративно-состязательные нейронные сети в задачах определения трендов // Московский экономический журнал. 2019. № 6. С. 180–191. DOI: 10.24411/2413-046X-2019-16031

16. Будыльский Д.В. Применение рекуррентных нейронных сетей в задачах обработки текстов на естественном языке // Вопросы науки. 2015. Т. 6. С. 8–12.

17. Данченко В.В. Обзор средств разработки информационной системы, основанной на анализе восприятия текста // Информатика и прикладная математика. 2020. № 26. С. 31–34.

Information about the authors

Elena V. Bunova, Cand. Sci. (Eng.), Ass. Prof. of the Department of Applied Mathematics and Programming, South Ural State University, Chelyabinsk, Russia; bunovaev@susu.ru.

Vlada S. Serova, Master's student of the Department of Applied Mathematics and Programming, South Ural State University, Chelyabinsk, Russia; vladislava.serova.98@gmail.com.

Информация об авторах

Бунова Елена Вячеславовна, канд. техн. наук, доц. кафедры прикладной математики и программирования, Южно-Уральский государственный университет, Челябинск, Россия; bunovaev@susu.ru.

Серова Влада Сергеевна, магистрант кафедры прикладной математики и программирования, Южно-Уральский государственный университет, Челябинск, Россия; vladislava.serova.98@gmail.com.

The article was submitted 13.03.2022
Статья поступила в редакцию 13.03.2022