Named Entity Recognition for Information Security Domain

© I.A. Mazharov © B.V. Dobrov
Lomonosov Moscow State University
Faculty of Computational Mathematics and Cybernetics,
Moscow, Russia
ivanmazharov@gmail.com dobrov_bv@mail.ru

Abstract. This work is devoted to the research of methods of named entity recognition for texts in Russian. Two methods of extracting information based on artificial neural networks were implemented, which were then tested on the collections of FactRuEval and Person 1000. The result of this work was the application of implemented software systems to a collection of texts on the topic of information security and analysis of the results.

Keywords: Named entities, information security, neural networks.

1 Introduction

Named entity recognition (hereinafter NER) is one of the most frequent tasks of processing natural language. The goal of NER is to find certain words and phrases in the text and classify them according to predefined categories (hereinafter labels), such as people's names, names of geographical objects, names of organizations, expressions of time, quantity and so on. The selected entities can be further used in applications for extracting information from the text. They can also be used as extracted properties for other natural language processing tasks.

Named entity recognition is an important source of information for various systems for extracting information and processing texts in natural languages. Possible applications of NER are search engines, cross-language information retrieval and machine translation, automated news gathering, question-answer systems, information retrieval for natural language processing systems, medical texts analysis. In addition, named entities are an important resource for structuring text data, which can help to extend text collections.

Named entity recognition associated with information security (hereinafter IS) will help in time to detect the emergence of a new threat, a virus or vulnerability in the network and take appropriate protective measures. The number and types of extracted entities will help to conduct a temporary and quantitative analysis of publications on the topic of information security, identify weaknesses and vulnerabilities of systems, and this will help with finding a solution to the problem.

In connection with the increased frequency of cyberattacks, as well as the consequent increase in the number of sources of unstructured or weakly structured information on the topic of information security, together with the general attention to this topic, make it urgent for research and further application in this perspective of the task of extracting named entities.

2 Approaches to solving the task of named entity recognition

Early NER systems were based on a set of rules defined manually. This approach used search and recognition by grammatical and syntactic patterns, according to the structure of the language in which the text is written. In this case, it is not necessary to have a large collection of marked data, but the drawbacks of this approach include a poor ability to generalize (addition of a new entity or change of the language will inflict reworking most of the rules) and the inability to learn by examples. The development of such systems takes a long time, and without significant processing, they cannot be applied to different types of texts or to different languages.

To solve these problems, algorithms have been developed for extracting named entities based on machine learning with different types of training: supervised learning, semi-supervised learning, unsupervised learning and reinforcement learning. Supervised learning is the most studied [1] and includes the method of support vector networks (SVM) [2], models based on the principle of maximum entropy [3], the method of decision trees [4] and the methods of matching the labels of a sequence of words, for example, the hidden Markov model [5], the Markov model of maximum entropy [6], the Markov model of conditional random fields (CRF) [7]. Like the approaches based on given rules, these methods rely on the features of the texts selected manually, which is a complex and time-consuming task, the result of which cannot be generalized to different data sets.

Recently, better results have been achieved using artificial neural networks, in comparison with other supervised learning algorithms for the task of extracting named entities [8], [9]. The advantage of neural networks lies in their ability to automatically take into account and translate syntactic and grammatical data into an internal representation and learn the model parameters according to the initial data set rather than rely on the signs highlighted by rules manually created for specific data.

Proceedings of the XX International Conference “Data Analytics and Management in Data Intensive Domains” (DAMDID/RCDL’2018), Moscow, Russia, October 9-12, 2018

Table 1 - statistics of marked corps

Corpus	**Tokens**	**Words and numbers**	**Unique tokens**	**PER**	**LOC**	**ORG**	**O**
FactRuEval 2016	90313	73833	22358	3350	2531	3324	81108
Person-1000	343023	300378	43802	27989	-	-	315034
Person-1111	297669	255374	47464	12056	-	-	285613

Table 2 - statistics of corpus on the topic of information security

Corpus	**Tokens**	**Words and numbers**	**Unique tokens**	**PER**	**LOC**	**ORG**	**O**	**POST**
Sec_col	377364	318466	53212	2950	2041	8670	345946	135
	TECH	**DEVICE**	**PROGRAM**	**EVENT**	**MEDIA**	**HACKER**		
	4378	832	7327	899	222	18		
	MISC	**VIRUS**	**GEOPOLIT**	**HACKER_GROUP**		**ARTEFACT**		
	148	442	6	79		3271		

For these reasons, systems of this type can be applied to different languages without significant architectural changes.

Evaluation of the recognition system for named entities is a way to test its operation and is performance on a manually marked set of data. The named entity is defined by its boundaries (consecutive words of one essence) and its type.

At the CoNLL conference, the following evaluation method was proposed: if the type and boundaries of the named entity, determined by the system, coincide with the type and boundaries of the selected experts, then the entity is considered to be properly allocated, otherwise the entity is not marked correctly. This method is called the exact (full) matching method.

Quality indicators of the NER system are Recall, Precision and F-measure (hereinafter R, P and F respectively), which are calculated as follows:

$$P = \frac{number\ of\ correctly\ extracted\ entities}{number\ of\ all\ extracted\ entities}$$

$$R = \frac{number\ of\ correctly\ extracted\ entities}{number\ of\ all\ entities}$$

$$F = \frac{2 * P * R}{P + R}$$

3 Formulation of the problem

In the framework of this thesis, it was required to develop and evaluate methods for extracting named entities in texts on the topic of information security using artificial neural networks. Modern methods applied to the task of extracting named entities are mainly related to various methods of machine learning.

The task described above is divided into the following sub tasks:

- Conduct quality testing of the developed methods at the collection Dialog Evaluation 2016, Persons-1000 and Persons-1111
- Apply developed software systems to a collection of texts on information security topics.

In order to train and test the implemented systems, the marked corpuses were used. At the moment, there are only a few corpuses for the NER task in Russian. In this paper, experiments were carried out with the following corpuses:

- FactRuEval 2016 corpus contains news and analysis materials collected on the resources of Private Correspondent and Wikinews, which are marked with following labels: PER, LOC, ORG and O (hereinafter Person, Location, Organization and None respectively). Subjects of the texts are political.
- The Person-1000 corpus [10] contains Russian news texts with marked named entities such as PER.
- The Person-1111 corpus contains Russian news texts with marked named entities such as PER.
- Security_collection corpus (provided by MSU Research Computing Center) contains texts on information security, marked with the help of the "Brat" system

Statistics for these corpuses are presented in Table 1 and Table 2.

4 Methods for solving the task of named entity recognition using artificial neural networks.

At the moment, the most widely used approaches to solving the problem of NER using artificial neural networks, which are divided into two large types: fully connected / convolutional neural networks and recurrent neural networks (hereinafter RNN). Recurrent networks allow you to store in memory and correlate various elements of the sequence, which enables them to show better results than full-connected / convolutional networks, in which the connection between words is established by passing not by words, but by word groups (windows). Currently, the standard solution for extracting named entities for English, German, Dutch, Spanish [9], [11] and Russian [12] languages is achieved using hybrid models combining Bi-LSTM and CRF. In this paper, both approaches will be considered and applied for the NER task for texts on the topic of information security in Russian.

4.1 Fully-connected neural networks in the NER problem

The first approach to the task of extracting named entities is based on a fully connected neural network [8]. It is based on the idea that the properties extracted from the sentence, after minimal processing, will be transferred to the input of a multilayer neural network, trained by the backpropagation method. The system will accept the input sentence and train several layers of recognition of properties that process the input data. The next layers of the neural network analyzing the properties of the sentence will be automatically trained to fully correspond to the task.

The system implemented in the framework of this work is based on the neural network proposed in [8], which is schematically presented in Figure 1. The first layer extracts the properties of each word. The second layer extracts the properties of the "window" of words, treating it as a certain sequence with internal and external structure (i.e., not treating it as a "bag of words"). The following layers are standard layers of the neural network - linear layers and the activation layer.

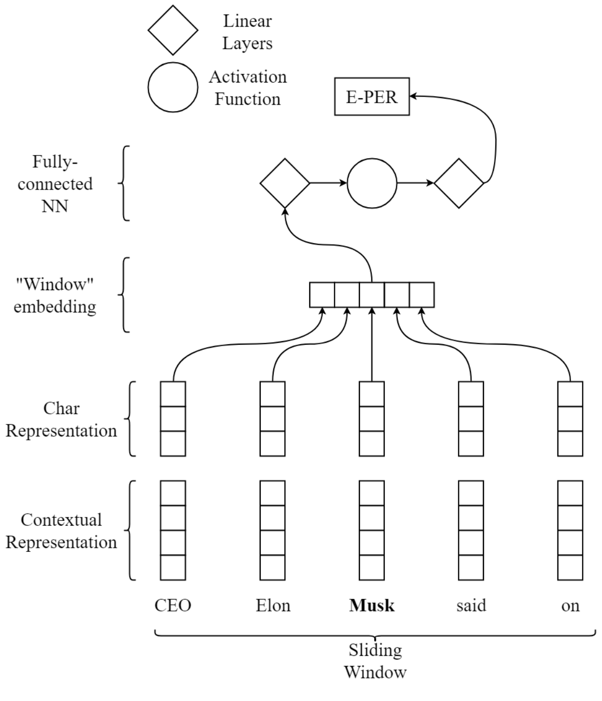

Figure 1 - fully connected neural network with a "window" approach

4.2 Recurrent neural networks in the NER problem

Recurrent neural networks are a powerful family of connected models that capture and analyze temporal changes through cycles in a graph. In theory, such networks can support the storage and transfer of dependencies on long sequences, but in practice, because of fading / explosion of gradients in the reverse propagation of errors, such dependencies are lost [13].

4.2.1 Networks of long short-term memory (LSTM)

The network of long-term short-term memory (LSTM) [14] is a variant of recurrent networks aimed at solving the gradient attenuation problem. In general, the cell of long short-term memory consists of three multiplicative gates that control the proportion of information that must be forgotten or passed on to the next step.

Below are the expressions for calculating these components:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i)$$
$$f_t = \sigma\left(W_f h_{t-1} + U_f x_f + b_f\right)$$
$$\tilde{c}_t = tanh(W_c h_{t-1} + U_c x_t + b_c)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o)$$
$$h_t = o_t \odot \tanh(c_t)$$

Where σ is the elementwise sigmoid function and $\odot$ is element wise multiplication. x_t is the input vector (for example, the weight of the word) at time t, and h_t is the vector of the hidden state (also the output vector) in which all useful information is stored at (and before) the time t. U_i, U_f, U_c, U_o denote the weights of the matrices of the various gates for the input data x_t, and W_i, W_f, W_c, W_o are the weights of the matrices for the hidden state. b_i, b_f, b_c, b_o denote the bias weights.

4.2.2 Bi-directional networks of long short-term memory (Bi-LSTM)

For the task of named entities recognition for a word, it is important to consider the past (left) context and the future (right) context. However, the hidden state vector h_t stores information only about the past, but not about the future. An elegant solution with proven effectiveness are bidirectional networks of long short-term memory [15]. The basic idea is that on the forward and backward pass of the sequence, two vectors of the concealed state are formed to take into account both the future and the past. Which then are combined into one common vector of output data.

4.2.3 Conditional random fields (CRF)

For the task of extracting named entities, it is also important to take into account the links between the labels of words that are located side by side. Moreover, to decode the result of the evaluation for the label follows in the perspective of the whole sentence, because, for example, next to the name will often be a surname and the like. Thus, it is necessary to apply conditional random fields, which will decode the sequence of words (sentence) and not every word individually.

Formally, if for the input sentence $X = (x_1, x_2, \dots, x_n)$ designate a matrix P of size $n \times k$, where k is the number of different labels, for the matrix of estimates of this sentence, where $P_{i,j}$ is responsible for the probability of the j-th label for the i-th word, then for the series of predictions $y = (y_1, y_2, \dots, y_n)$, its estimate can be determined as

$$s(X, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

Where A is the matrix of the probable transition, in which $A_{i,j}$ denotes the transition probability from the label i to the label j. Then the conditional probability function in the completely possible way along the labels for the sequence y will be expressed using the softmax function: $p(y|X) = \frac{e^{s(X,y)}}{\sum_{\tilde{y} \in Y_X} e^{s(X,\tilde{y})}}$

During the training the log likelihood function is maximized

$$\log(p(y|X)) = s(X, y) - \log(\sum_{\tilde{y} \in Y_X} e^{s(X,\tilde{y})})$$

Where Y_X denotes all possible label sequences for the sentence X.

4.2.4 Vector representation of words

For each word, it is necessary to obtain its vector representation $\omega \in \mathbb{R}^n$, which will be relevant for the NER task. It can be considered as a concatenation of the weights of a word from the pre-trained model $\omega_1 \in \mathbb{R}^{d_1}$ and the property vector $\omega_2 \in \mathbb{R}^{d_2}$, obtained from the letter representation of the word. In this work, a bidirectional network of long short-term memory will be used to extract attributes from the letter representation of the word, which will be fed to the input with words in the form of a sequence of letters; its architecture is depicted in Figure 2. However, another approach is also possible based on other recurrent or convolutional networks [9] [16] [17].

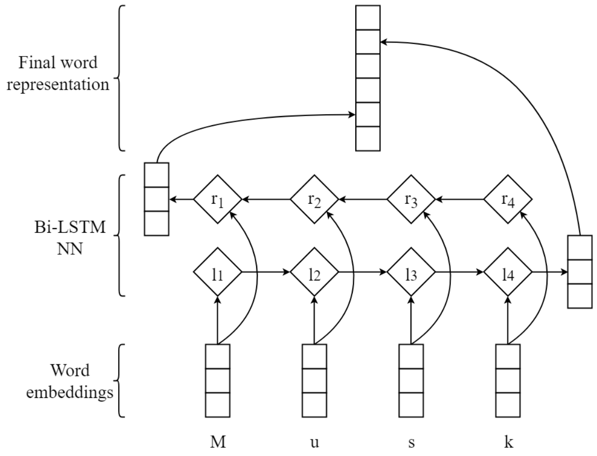

Figure 2 - Bi-LSTM character level architecture

4.2.5 Bi-LSTM+CRF

To obtain labels for the proposal, the Bi-LSTM network is also used, its architecture is depicted in Figure 3, and its further training is applied to receive word weights. Whose results will be transferred to the CRF model, which was described above.

5. Neural networks training

Models for neural networks were implemented in Python using the Keras and Tensorflow libraries. The training was conducted on a video card GeForce GTX 1080. One epoch took 370 seconds for a fully connected network and 190 seconds for Bi-LSTM. The training lasted for 30 epochs.

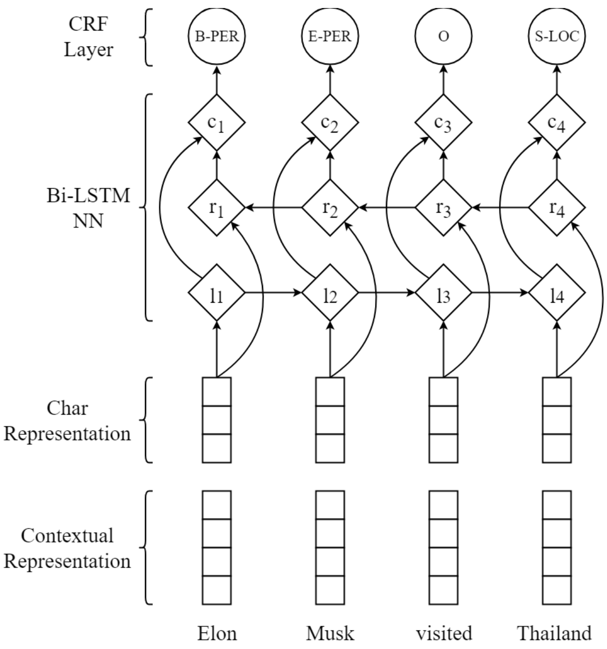

Figure 3 - the architecture of the Bi-LSTM word-level network

5.1 Parameters

Weights of words. To obtain a vector representation of words, the results of the Russian Distributional Thesaurus project [18] were used. This corpus contains 12.9 billion tokens, extracted from the collection of books in Russian. For the construction of vector representations of words, the standard implementation of word2vec was used with the parameters given below. These parameters allow achieving the best results from the point of view of the quality assessment for several test collections [19].

- Model: skip-gram
- Context window size: 7 words
- Dimension of vector space: 500
- Number of iterations: 5
- Minimum word frequency in the case: 5

In a system with a fully connected neural network, for each word, a vector is first extracted from the word2vec model of dimension 500. If the word is not present in this model, the vector of the weights is initialized by random values over a uniform distribution.

In addition, experiments were conducted with the Araneum Russicum Maximum model of the project RusVectores, which was assembled by V. Benko in 2016.

Table 3 - results of testing on standard corpuses

Model		**Fully connected NN**	**Bi-LSTM**	**Bi-LSTM + CRF**		
Corpus		**FactRuEval**	**FactRuEval**	**FactRuEval**	**Person-1000**	**Person-1111**
PER[17]	**P**	89.69	89.85	94.98	92.12	93.85
	R	94.62	92.33	97.48	92.85	94.26
	F	92.08	91.07	96.21	92.6	94.05
LOC[1]	**P**	88.18	88.97	93.88	-	-
	R	85.71	87.55	92.86	-	-
	F	86.93	88.25	93.36	-	-
ORG[1]	**P**	78	83.56	88.86	-	-
	R	68.36	75.24	79.85	-	-
	F	72.86	79.18	84.11	-	-
Total[18]	**P**	78.18	84.45	89.21	90.25	90.95
	R	83.11	86.59	91.41	91.01	91.21
	F	80.57	85.51	90.29	90.63	91.08

Table 4 - comparison by type PERSON

Model	**Person 1000 (only PERSON)**			**FactRuEval 2016 (only PERSON)**		
	P	**R**	**F**	**P**	**R**	**F**
Trofimov	0.97	0.93	0.95	-	-	-
Mozharova at al.	-	-	0.97	-	-	-
Grey	-	-	-	0.96	0.87	0.91
Violet	-	-	-	0.94	0.92	0.93
FC NN	-	-	-	0.89	0.94	0.92
Bi-LSTM+CRF	0.92	0.92	0.92	0.94	0.97	0.96

This corpus contains about 10 billion words, which are represented by vectors of dimension 100.

Character weights. For a model with a fully connected neural network, the features of the word level were extracted with the help of given rules and their training was not carried out. For the Bi-LSTM model, the weights of the letters were initialized using the Xavier [20] method and had a dimension of 100.

Additional parameters. In order to improve the results for a system with a recurrent neural network, for each word external features were added - the presence or absence of a word in specialized dictionaries of named entities on the topic of information security. A total of 12 dictionaries were used, with such named entities as virus names, program names, etc., with the total number of tokens in 1470. Since the words in the dictionaries were brought to normal form, all the words of the corpus were also reduced to normal form using the morphological analyzer pymorphy2. For the Bi-LSTM + CRF + voc model, the feature vector is constructed as a concatenation of the vector of pre-weighed word weights and a binary vector that carries information about those dictionaries in which the word was encountered. The intersection of dictionaries with the corpus amounted to 576 tokens, which is ~ 0.15% of all corpus tokens.

5.2 Optimization algorithm

As an optimization algorithm, the Adam [21] algorithm was chosen. The initial rate of training was equal to $\mu_0 = 0.005$ with decreasing in each epoch according to the linear law $\mu_t = \mu_0 * p * t$, where the decay factor was equal to $p = 0.9$, and t – the number of passed epochs. In addition, experiments were conducted with optimization algorithms for SGD and AdaGrad, but these methods showed no improvement in comparison with Adam and converged more slowly.

To overcome the retraining, a Dropout method with a probability of 0.5 was used, which gave a significant increase in the accuracy of the model.

To overcome the attenuation of the gradients, the gradient equalization method was used.

6. Evaluation

6.1 The standard task of NER

The purpose of the first experiment was to verify that the implemented systems are correct and their application for texts on the topic of information security is justified. For this, tests were carried out on three corpuses: the

---

[17] Incomplete matching

[18] Full match

corpus of FactRuEval 2016, the corpus of Person-1000 and the corpus of Person-1111. The results are shown in Table 3. For the first two corpuses, comparisons are made with the known results achieved by the PERSON type in Table 4.

Table 5 - results of testing on standard corps

Model	P		R		F	
	Full.[2]	Incomp.[1]	Full.	Incomp.	Full.	Incomp.
Fully connected NN	60.91	64.85	56.93	58.49	58.85	61.51
Bi-LSTM + CRF	70.33	76.12	70.44	71.24	70.39	72.98
Bi-LSTM + CRF + vocab	71.22	76.19	71.01	71.42	71.11	73.25

As can be seen, adding a CRF layer significantly improves the quality of predictions. Moreover, Bi-LSTM networks outperform the fully connected in the task of allocating named entities. In addition, the experiment showed that the implemented methods are highly rated by the F1 measure and their use in the task of named entities recognition for texts on information security is justified.

6.2 The task of NER on the topic of information security

The purpose of the second experiment was to establish the applicability of current solutions to the NER problem for IS texts. In this task, it was suggested to train systems to extract from the text 16 types of named entities.

Test results are shown in Table 5. As can be seen from the table, the results fell for all implemented solutions, however, the use of specialized dictionaries allowed to improve the result.

6.3 Additional parameters

Weights of words. Table 6 compares the results for the two different models of word weights discussed earlier. RDT stands for Russian Distributional Thesaurus and ARM for Araneum Russicum Maximum correspondingly. It can be seen from the table that a model with a large number of weights gives better results.

Table 6 - comparison of pre-trained models

Model	P	R	F
RDT	70.33	70.44	70.39
ARM	69.02	69.13	69.07

Table 7 - comparison of the Dropout layer parameter

Dropout	P	R	F
-	70.23	70.34	70.29
0.4	71.00	70.30	70.65
0.5	71.47	70.35	70.91
0.7	69.88	68.64	69.25

Table 8 - comparison of the gradient clipping parameter

Clipping	P	R	F
-	70.79	69.97	70.38
10.0	71.28	70.87	71.07

Dropout. Table 7 compares the results with the addition of the Dropout layer and without it, as well as the various probabilities of using Dropout. It can be seen from the table that the addition of this layer improved the results of the system, and the optimal value was the probability of 0.5.

Gradient clipping. Table 8 compares the results for different values of the gradient alignment parameter.

Normalization of words. Experiments were carried out in which the vector of its norm was put in correspondence for each word. This approach showed little improvement, but it takes a lot of time to bring the words to the initial form.

6.4 Analysis of results

Despite the fact that the values of the metrics on the corpus for information security were lower than on the corpuses with standard named entities, this result allows us to state that the task of NER in IS texts is more complex and requires the development of new methods and approaches.

Above, the method of adding dictionaries to the model has already been considered, now it is suggested to consider the changes made to it in the metrics by the relevant entities, namely HACKER, HACKER_GROUP, VIRUS, DEVICE, TECH, and PROGRAM. The results are shown in Table 9. All metrics are calculated by incomplete matching.

It can be seen from the table that the improvement occurred almost in all relevant problems to entities. For some types of F1 measures showed significant growth, for the type HACKER_GROUP the difference was more than 150%. Thus, despite the fact that the size of the dictionaries was small in comparison with the size of the corpus (about 0.15%), one can see that the use of thematic dictionaries correlated with the task positively affects the F1 measure.

The following reasons for the decrease in the accuracy of the allocation were also highlighted:

- Increase in the number of entity types. This undoubtedly leads to a decrease in the quality of the system, as the number of types increases, the complexity of isolating a specific one grows, and new dependencies between new types appear, which the system cannot simply take into account. Also, the extracted signs become more sensitive.
- Semantic proximity of entity types. This factor makes itself felt even at the stage of marking the corpus by

assessors. Ambiguities in markup appear already at the stage of four types (that is why sometimes the type LOCORG mentioned earlier was introduced) and they become even larger, with the number of entity types increasing and their semantic convergence, which generates errors even before learning the named entities recognition system.

Table 9 - test results by types of relevant entities

Model	**Bi-LSTM + CRF**			**Bi-LSTM + CRF + vocab**			**Total amount**
	P	**R**	**F**	**P**	**R**	**F**	
HACKER	100	25	40	100	25	40	4
HACKER_GROUP	100	28.57	44.44	100	64.29	78.26	28
VIRUS	74.26	56.39	64.10	84.71	54.14	66.06	133
DEVICE	79.63	54.89	64.99	78.53	59.15	67.48	235
TECH	70.07	73.75	71.74	68.93	72.44	70.64	1036
PROGRAM	70.89	76.46	73.57	70.77	79.29	74.79	1487

- Heterogeneity of the marked corpus. Since the marked data were not scientific articles, the news was not written in a formal language, there are many stylistic, lexical and spelling mistakes, many borrowings, English text, jargon and common speech. All this negatively affected the quality of the NER system, including, for many words, no correspondence was found in the vector representation of word2vec. For the information security corps, this figure was 10044 out of 48,320 words (20.7%), while 1021 out of 20,908 words (4.8%) were not found for the corpus of FactRuEval, respectively.
- Low degree of occupancy in classes that are relevant to the topic of information security. For almost all classes, the share in the total number of words of the corps was significantly less than 1%, while in standard corpuses this share does not fall below 3.5-4% of the total number of words in the corpus.

7 Results

Within the framework of this work, the task is to extract named entities from texts on the topic of information security from the use of artificial neural networks. The novelty of the study is that at the moment there are no known publications on the named entities recognition from the texts on information security in Russian. In that work:

- Implemented two software systems based on artificial neural networks, solving the problem of named entity recognition.
- The developed systems were tested on the existing marked packages in Russian using the standard for the task of extracting named entities: PERSON, LOCATION, and ORGANIZATION. Thus, it was confirmed compliance with the modern level of quality for such systems.
- The developed software systems were transferred to the corpus on the topic of information security. The recognition was performed on fifteen types of named entities. Based on the results of testing on this corpus, the following quality indicators were obtained for the main classes relevant to the topic of information security: PROGRAM - F1-measure 73.57%; TECH - F1 measure 71.74%; DEVICE - F1-measure 64.99%; VIRUS - F1-measure of 64.10%; HACKER_GROUP - F1-measure of 44.44%; HACKER - F1-measure 40%.
- It was found out that the use of small, specialized dictionaries of named entities improves the quality indicators for all classes by 0.5-1% and 3% for classes relevant to the topic of information security.

Area of information security is complex and already the methods studied do not achieve results in this area that are close to the results in standard NER tasks, this is a consequence of the following factors: an increase in the classes of named entities, difficult manual classification, semantic proximity classes, lack of representativeness of the classes in marked corpuses.

References

[1] Nadeau, David and Satoshi Sekine (2007) A survey of named entity recognition and classification. Linguisticae Investigationes 30(1):3–26.

[2] Masayuki Asahara and Yuji Matsumoto. 2003. Japanese named entity extraction with redundant morphological analysis. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1. Pages 8-15

[3] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. 1998. NYU: Description of the MENE named entity system as used in MUC-7. In Proceedings of the Seventh Message Understanding Conference (MUC-7).

[4] Satoshi Sekine et al. 1998. NYU: Description of the Japanese NE system used for MET-2. In Proceedings of the Seventh Message Understanding Conference (MUC-7).

[5] Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. Proceedings of the fifth conference on

Applied natural language processing. Pages 194-201

[6] Kumar Saha, Sujan, Sarathi Ghosh, Partha, Sarkar, Sudeshna, & Mitra, Pabitra. (2008). Named Entity Recognition in Hindi using Maximum Entropy and Transliteration. Polibits, (38), 33-41

[7] Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4. Pages 188-191

[8] Ronan Collobert, Jason Weston, L´eon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. The Journal of Machine Learning Research Volume 12, 2/1/2011 Pages 2493-2537

[9] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 .

[10] Vlasova N.A., Suleymanova E.A., Trofimov I.V: Report on Russian corpus for personal name retrieval. In proceedings of computational and cognitive linguistics TEL’2014, Kazan, Russia, pp 36 – 40 (2014)

[11] Xuezhe Ma and Eduard Hovy. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics 1(Long Papers):1064-1074.

[12] L. T. Anh, M. Y. Arkhipov, M. S. Burtsev. 2017. Application of a Hybrid Bi-LSTM-CRF model to the task of Russian Named Entity Recognition. arXiv preprint arXiv:1709.09686 .

[13] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166.

[14] Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. Neural computation, 9(8):1735–1780.

[15] Sepp Hochreiter, Jrgen Schmidhuber: Long Short-Term Memory. MIT Press, Vol.9, No. 8, 1735-1780 (1997).

[16] Chiu, J.P., Nichols, E.: Named entity recognition with bidirectional lstm-cnns. arXiv preprint arXiv:1511.08308 (2015)

[17] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O’Reilly Media Inc.

[18] Arefyev N., Panchenko A., Lukanin A., Lesota O., Romanov P. (2015): Evaluating Three Corpus-Based Semantic Similarity Systems for Russian. In Proceedings of the 21st International Conference on Computational Linguistics and Intellectual Technologies (Dialogue'2015). Moscow, Russia. RGGU

[19] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python, Pedregosa et al., Journal of Machine Learning Research 12, pp. 2825-2830.

[20] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, 2010.

[21] Diederik P. Kingma, Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization arXiv preprint arXiv:1412.6980