

Neural Approach for Named Entity Recognition

Kateryna Yalova^a Kseniia Yashyna^a and Iryna Ivanochko^b

^a *Dniprovsk State Technical University, Dniprobudivska str.2, Kamyanske, 51918, Ukraine*

^b *University of Vienna, Universitätsring 1, Vienna, 1010, Austria*

Abstract

The work presents the results of bidirectional long short term memory (BiLSTM) neural network with conditional random fields (CRF) architecture for named entity recognition (NER) problem solving. NER is one of the natural language processing (NLP) tasks. The NER solution allows to recognize and identify specific entities that are relevant for searching in particular data domain. The generalized NER algorithm and neural approach for NER with BiLSTM-CRF model are presented. The use of CRF is responsible for prediction the appearance of searched named entities and improves the recognition quality indicators. The result of the neural network processing is input text information with recognized and designated named entities. It is proposed to use weakly structured resume text information to conduct experiments with BiLSTM-CRF model for named entities recognition. Ten types of named entities are chosen for neural network processing, such as: person, date, location, organization, etc. Own created corpus of resume documents marked manually was used as a data set for BiLSTM-CRF neural model training, validation and testing. Analysis of the adequacy of the proposed approach was carried out using precision, recall and balanced measure F1 metrics. The average recognition values on the testing set were: precision 79,06%, recall 71,51% and F1 75,09%. The best recognition scores were obtained for named entity “date”: precision 92,12%, recall 81,60%, F1 86,54%. The developed neural model and software have practical value for solving problem of resume summarizing and ranking candidates for work as they can be used to form an array of incoming data.

Keywords

Neural network, BiLSTM, CRF, named entity recognition

1. Introduction

Named entity recognition is one of the many tasks of natural language processing – a generalized trend of artificial intelligence and mathematical linguistics, which explores the problems of computer analysis and synthesis of natural languages [1]. A named entity is a sequence of words that can be assigned to a specific category. The problem of named entity recognition involves searching, selecting certain continuous fragments in the input text, and correlating the found fragments with an established set of named entities. A description of the task of searching for named entities was first presented in 1996 at the Sixth Message Understanding Conference (MUC-6) [2]. The formulated task involved finding and identifying six entities in the text: names of persons, names of organizations, geographical names, dates, monetary amounts, and percentage values. Identification of these entities in the text was recognized as one of the most important subtasks of natural language processing. After setting the NER problem, the idea of searching for certain entities was transferred to various data domains and languages. To date, NER is used for texts of various topics, and the set of named entities itself was proposed to be formed taking into account the functional requirements of each specific task. The most

IntelITSIS'2021: 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 24–26, 2021, Khmelnytskyi, Ukraine

EMAIL: yalovakateryna@gmail.com (K. Yalova); yashinaksenia85@gmail.com (K. Yashyna); iryna.ivanochko@univie.ac.at (I. Ivanochko)

ORCID: 0000-0002-2687-5863 (K. Yalova); 0000-0002-8817-8609 (K. Yashyna); 0000-0002-1936-968X (I. Ivanochko)



© 2021 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

popular application fields of NER are event recognition from news, patient data extraction from medical documents, publication analysis on social networks.

2. Related works

Many methods are used to solve the NER problem, however, methods using neural networks and machine learning show better results for various languages and text document corpus [1]. In the works [3-4], the neural network approach to NER is used to search for named entities in the field of medicine, such as disease, drugs, and symptoms, appearing in publications on the Twitter social network [3] and obtained from medical records of the Spanish Meddocan system [4]. And in the work [5], the named entities hacker, hacker group, program, virus, device, etc. are searched for to solve the problem of data labeling and extraction in the field of cybersecurity.

In the works [6-9], the authors justify the feasibility of using BiLSTM-CRF neural networks to solve the NER problem by comparing the recognition results using different neural network architectures: Recurrent Neural Networks (RNN), Document Context Language (DCLRNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), LSTM with Conditional Random Fields (CRF), BiLSTM-CRF. The quality of recognition largely depends on the data set that is used for training, validation, and testing the neural model. In studies that use prepared training document corpora, such as Switchboard Dialog Act Corpus (SWDA) [7], Sec_col, BioCreative V CRD corpus [8], FactRuEval 2016, Gareev's dataset, Person 1000 [10], Wiki-727, Choi, RST-DT [11-12], People's Daily data [13] – recognition accuracy is on average 90%. The recognition quality obtained on data sets formed and labeled by the authors manually [5, 14-15] is significantly lower – on average 60-80%, and for some named entities it is about 30%.

The main purpose of this paper is to present the results of the neural network implementation for NER and the quality evaluation of the proposed solutions on the example of low-structured resume information. To achieve this purpose, the following tasks were completed:

- Defining a set of named entities and forming an incoming document corpus;
- Labeling a text corpus with a set of named entities and preprocessing of incoming data;
- Development, implementation, and training of the BiLSTM-CRF neural network model;
- Testing and evaluating the quality of NER.

The choice of a resume as a dataset for NER is justified by the fact that the texts of various resumes can be found in the public domain on job search websites, they are created in a weakly structured form suitable for solving the NER problem. Despite the fact that a resume document is only 2-3 pages of text information, such documents are full of dates, names of organizations and locations, and may contain a lot of additional information – information noises that are irrelevant to the purpose of recognition. In addition, the solution of the NER problem on the resume document corpus is of practical value, since the resulting recognition data allow to automatically form a database of job candidates. The proposed solutions and the developed software are justified for use in data domains with a large mass of incoming information flows, for example, for recruitment agencies.

3. Models, methods and technology

For implementation by the natural language processing mechanism and named entity search, various machine learning methods are used to find complex patterns in incoming data sets. The analysis of the scientific literature shows the prospects of using BiLSTM for the NER problem.

3.1. BiLSTM-CRF neural network

A BiLSTM network is a neural network that is built on the principles of LSTM. LSTM is a special architecture of the Recurrent Neural Network, which is able to study long-term dependencies, proposed in 1997 by S. Hochreiter and Y. Bengio. LSTM has the form of a chain of repeating modules, the cell of which is able to calculate the current hidden state h_t based on the current vector x_t , the previous hidden state h_{t-1} , and the previous state of the cell c_{t-1} . LSTM consists of c – memory cell

and three cell gates: i – input gate, o – output gate, f – forget gate, that have the same size as hidden vector h and are calculated as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \quad (1)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f), \quad (2)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o), \quad (4)$$

$$h_t = o_t \tanh(c_t), \quad (5)$$

where σ – the sigmoid function element, W_{xi} , W_{xf} , W_{xo} , W_{xc} – the weights between current vector and gate vectors; W_{hi} , W_{hf} , W_{ho} , W_{hc} – the weights between hidden vector and gate vectors; W_{ci} , W_{cf} , W_{co} – the weights between cells and gate vectors, that are diagonal, so element k in each gate vector only receives input from element k of the cell vector [6]; b – biases.

The forget gate is a sigmoid layer that decides what information does not have to be saved in the memory cell. The input gate determines which values must be updated in the memory cell, and the tanh-layer builds a vector of new values of candidates that can be added to the memory cell. After updating the state of the memory cell, a decision is made about the output data. To do this, a sigmoid layer is applied to the memory cell, which decides what information to output, then the values from the memory cell pass through the tanh-layer so that only the required information is output.

The main difference between BiLSTM and LSTM is that in LSTM, for each vector x_t , the hidden state h_t receives only past information. Whereas the BiLSTM architecture allows taking into account both the forward context \overrightarrow{h}_t and the backward \overleftarrow{h}_t context by concatenating them. In this case, the left context is calculated first, then the right context is calculated in the opposite direction, after which the results of these actions are combined to form a complete representation for the input sequence element [16]:

$$h_t = \overrightarrow{h}_t \oplus \overleftarrow{h}_t. \quad (6)$$

A characteristic feature of the BiLSTM architecture is that it is able to learn long-term dependencies and has minimal requirements for the learning process [13].

In the works [4, 6–8, 10–13, 16] the efficiency of adding a CRF layer to the BiLSTM architecture is proved to optimize the calculation of the probability of appearance of the searched named entities and to improve the quality indicators of NER in general. The CRF layer is a discriminative probabilistic model that takes into account the context of the object being classified and is used to predict sequences [17].

Let $x=(x_1, \dots, x_t)$ be the set of an incoming text sequence of length t , for example, a set of words in a sentence. Then $y=(y_1, \dots, y_t)$ is the set of corresponding labels (for example, $y_1 = \text{“Person”}$). The set x and the set y form the set of random variables $V=x \cup y$. Then, to solve the problem of correlating an element of the incoming sequence with a named entity, the conditional probability $p(y|x)$ needs to be determined.

The potential function is calculated as follows [18]:

$$\varphi(x_{\{k\}}) = \exp(\sum_k \lambda_k f_k(y_{t-1}, y_t, x, t)), \quad (7)$$

where $f_k(y_{t-1}, y_t, x, t)$ – an arbitrary feature function, which has following parameters: a label of the $t-1$ node, a label of the t -th node, an input set x , a position of predict node t , λ_k – a learned weight for each feature function, which algorithm will improve during processing.

The purpose of the feature function is to express a specific characteristic of the sequence represented by the data point. Each feature function is based on the label of the previous word and the current word and is either 0 or 1. To construct a conditional field, each function is assigned a set of weight factors λ_k . Then as a conditionally random field is considered a probability distribution of the following type:

$$p(y|x) = \frac{1}{z_x} \exp(\sum_t \sum_k \lambda_k f_k(y_{t-1}, y_t, x, t)), \quad (8)$$

where z_x – a normalization coefficient that can be found as:

$$Z(x) = \sum_y \exp(\sum_t \sum_k \lambda_k f_k(y_{t-1}, y_t, x, t)). \quad (9)$$

To use conditionally random fields, the necessary functions are first defined by initializing weights to random values, and then a gradient descent is applied iteratively until the parameter values (in this case, Lambda values) converge. Unlike other statistical methods, the CRF method requires a much

smaller training sample, since a statistically significant combination can be defined as a set of connected vertices for the object under study.

3.2. BiLSTM-CRF model application for the resume named entity recognition

The generalized NER algorithm, which assumes machine learning, consists of the following steps:

1. Setting the problem in terms of a specific data domain: selecting named entities and an array of data to search for them.
2. Input data pre-processing.
3. Creating a neural network of a specific architecture.
4. Training, validation, testing, optimization of hyper parameters of the developed neural network.
5. Using the trained neural network to solve the problem.

Figure 1 shows an overview of the information data flows from receiving input information to displaying specified named entities search results.

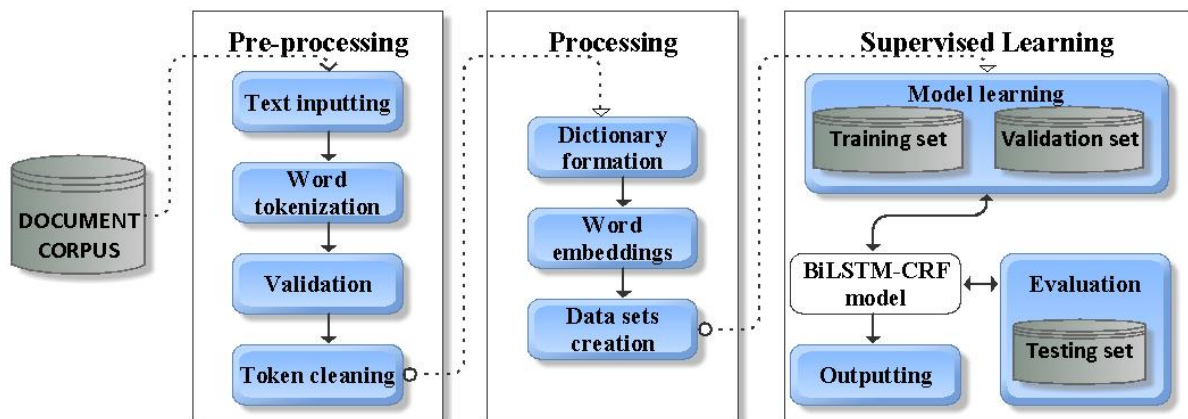


Figure 1: Generalized data flows scheme

The BiLSTM-CRF neural architecture (fig. 2) was tested on the example of solving the NER problem in a weakly structured text information of a resume.

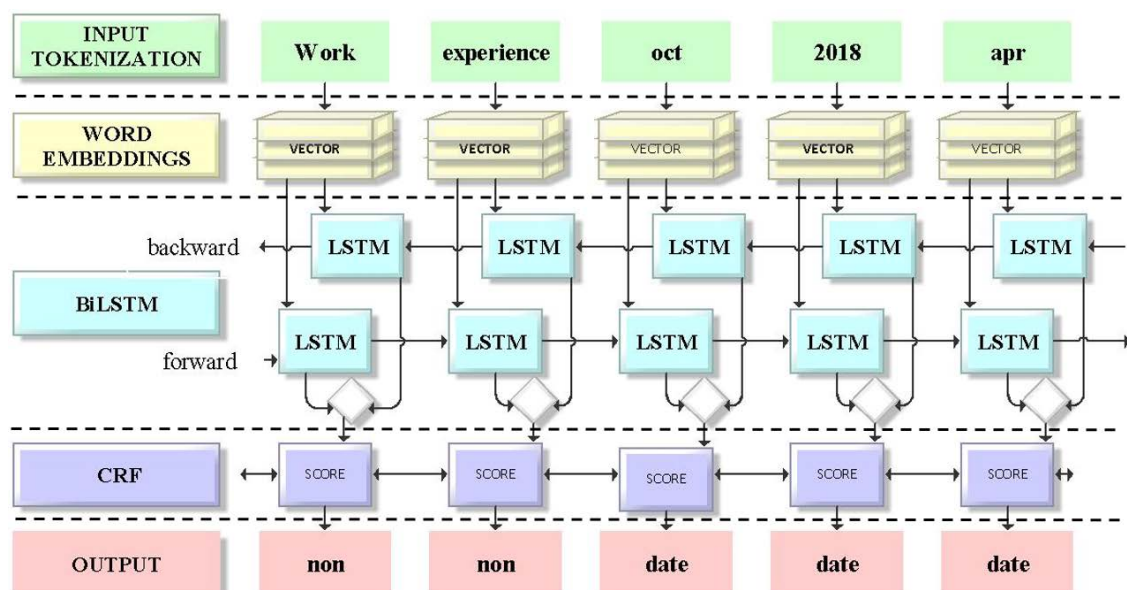


Figure 2: BiLSTM CRF neural network architecture

The figure 2 shows the architecture of the BiLSTM-CRF neural network with selected layers. After the input sequence has been included in the LSTM layer, probable estimates are calculated which reflect belonging to the label. Then these results are sent to the CRF layer, where these values are corrected. CRF decides which label the input value should be attributed to. The CRF layer contains a graph of the probabilities of transitions between labels that were acquired as a result of the training.

The developed software generates an array of incoming information for further resume summarizing, search, and ranking of suitable employees. As named entities in the data domain, 10 labels were selected: “Person” – data on the surnames, names, patronymics of people; “Phone” – phone numbers in the international format; “E-mail” – electronic mail addresses; “Date” – dates in various formats relating to the date of birth, dates of education and work; “Location” – names of geographical objects, such as locations of educational institutions and corporate employers; “Organization” – names of organizations, such as names of educational institutions or corporate employers; “Job title” – position names; “Education major” – titles of specialties and areas of education; “Education degree” – levels of education or qualification; “Job description” – description of the skills and professional competencies.

The main goal of the pre-processing stage of incoming data is to clear them, consolidate them, and transform them into a format suitable for transmission to a neural network. The algorithm for pre-processing data for modeling the operation of the BiLSTM-CRF neural model includes the following steps:

1. Formation of the corpus of resume documents.
2. Tokenization of the incoming sequence.
3. Word embedding.

The resume corpus was formed based on open data from job search websites and consists of 160 text resume files, labeled manually with selected named entities, used for training, validation, and testing of the BiLSTM-CRF model. The average number of characters in each resume is approximately 5000 characters, and the average number of words in one resume is up to 1000.

The pre-processing of incoming text information begins with tokenization – splitting the raw text into smaller blocks – tokens. Depending on the algorithms for splitting the incoming text into tokens, there are different methods of tokenization. In this paper, we use the Treebank Word Tokenizer algorithm [19], in which tokens are words defined on the basis of punctuation marks and white-space characters, and words with apostrophes and time periods are divided into their component parts. After tokenizing the incoming text, the received tokens are processed, which includes validation with regular expressions, for example, validation of hyperlinks, emails, phone numbers, dates, numbers, etc. Pre-defined characters, such as punctuation marks, defined as separate tokens by the Treebank Word Tokenizer algorithm, are also removed from the token list. Tokens that have passed validation are stored in a dictionary – a data structure of the key-value format, where the value is a token, and the key is a position of the token in the original text sequence.

In order to make the generated dictionary tokens suitable for transmission to a neural network, they must be converted to a vector form by implementing the word embedding. The word embedding allows matching an incoming word with a vector that displays its meaning in the space of semantic information. In 2013, T. Mikolov proposed an approach to the word embedding, which was called Word2Vec, it collects statistics on the co-occurrence of words in phrases, then uses neural network methods to solve the problem of reducing the dimension, and outputs compact vector representations. In this paper, the Skip-gram algorithm of the Word2Vec technology was applied. The Skip-gram algorithm uses the current word from the dictionary to predict the surrounding words. It was chosen because it is effective for small training sets [20]. In this paper, the dimension of the vector representations is $[300 \times n]$, where n is the number of values in the dictionary.

The resulting vector representations are fed to the input of the BiLSTM-CRF model, the hyperparameters of which are presented in Table 1.

Table 1
BiLSTM-CRF neural model hyper parameters

Hyper parameter	Values
Layers	2
Units	512
Epochs	24
Learning rate	0.001
Dropout	0.5
Mini batch mode	8

At the output, the model returns a projection on the input text, where each element of the sequence corresponds to the number of the named entity or is classified as a common text.

4. Experiment, Results and Discussions

The model was trained on a sample of data consisting of 100 resumes. The goal of the training was to reduce the loss function value and get a matrix of weights with which this value was obtained. Since in this paper, a relatively small training set was used to avoid over-training the neural model, the training phase was carried out using a training and validation data set. The resume corpus was divided into three non-overlapping subsamples of data: 100 – for training the model, 30 – for validation, and 30 – for testing. A validation set is an auxiliary set that is used not to change the network weights, but to determine the optimal values of hyperparameters. The use of the validation set makes it possible to determine the moment when the neural network begins to over-train and determine the number of training iterations – epochs, at which the value of the loss function will be minimal, and the neural network will produce the most accurate results [21]. In this paper, the following condition was determined: if the value of the loss function on the training data set decreases, but on the validation data set it remains unchanged or increases during n epochs, then this is the moment when the training of the system must be stopped. At $n=5$, the model was trained for 24 epochs. The training time was approximately 2 hours, taking into account that the incoming data set was divided into 8 parts, according to the number of processor cores of the computer on which the training was performed. After completing the training phase using the training and validation sets, a testing set was transferred to the model to determine the quality of the created model.

To assess the quality of named entity recognition the following metrics were used: precision (P) as a measure of quality, recall (R) as a measure of quantity, and $F1$ -score ($F1$) as a balanced feature of the model considering the values of P and R . P determines the ability of the model not to classify unnamed entities as named entities. R determines the ability of the model to correctly recognize named entities in the data set, regardless of whether irrelevant results also return. To determine the values of these metrics, the following concepts are introduced:

- True Positive (TP) – the number of tokens that are a specific named entity and are correctly recognized;
- True Negative (TN) – the number of tokens that are not a named entity and are correctly recognized;
- False Positive (FP) – the number of tokens that are not a named entity but are recognized as a named entity (incorrectly recognized);
- False Negative (FN) – the number of tokens that are a named entity but are not recognized.

Then the values of the metrics can be determined by the formulas [22]:

$$P = \frac{TP}{TP + FP} \quad (10)$$

$$R = \frac{TP}{TP + FN} \quad (11)$$

$$F1 = 2 \frac{P * R}{P + R} \quad (12)$$

For each resume from the test data set, the values of P , R , and $F1$ were calculated, the average values of these metrics were $P \approx 79\%$, $R \approx 71.5\%$, and $F1 \approx 75\%$. Table 2 presents the results of the quality evaluation of each named entity recognition on a test dataset.

Table 2
Results of named entities recognition quality evaluation

Named Entity	Testing		
	P	R	F1
Date	0,9212	0,8160	0,8654
Person	0,9136	0,6267	0,7434
Location	0,8500	0,7334	0,7874
Organization	0,8247	0,6740	0,7418
E-mail	0,7410	0,7190	0,7298
Education degree	0,8230	0,7856	0,8039
Edu_major	0,7572	0,7135	0,7347
Phone	0,8832	0,8129	0,8466
Job title	0,6400	0,6419	0,6409
Job description	0,5530	0,6287	0,5884

From the data presented in Table 2, it can be concluded that the obtained values of P , R , and $F1$ demonstrate the high ability of the proposed BiLSTM-CRF model to recognize selected named entities in a given data domain. The named entities Date, Person, Location, Organization are widely used, and the results of their recognition were compared with the works of authors who used the BiLSTM-CRF model on their own manually labeled text document corpora. The results of the comparison justify the adequacy of the proposed solutions.

The maximum values of P , R , and $F1$ are obtained for the named entities “Date” and “Person”, and the lowest values are obtained for “Job Title” and “Job Description”. The low recognition rates for “Job Title” and “Job Description” are related to the peculiarity of these entities – they are text data with weak semantic links to the surrounding context. In addition, it should be noted that for “Job Title” and “Job Description”, the recall value exceeds the precision value, which indicates that the model is able to recognize these named entities but poorly copes with finding differences of found named entities from the others.

The ways to improve the performance of the neural network with the unchanged BiLSTM-CRF architecture are the expansion of the corpus of resume documents used for training, validation, and testing of the network, the selection of hyperparameters, and the optimization of the pre-processing stage of incoming data.

To implement the described algorithms, the BiLSTM-CRF architecture, and the user software application, the Python programming language, the NLTK library, and PyTorch were used. To get NER results in a resume, the user must specify the path or file name to the resume text. At the output, the user receives the resume with found named entities marked with different colors. The obtained data are the basis for the implementation of automatic resume summarizing mechanisms, data analysis, resume ranking, and search for candidates with a different combination of incoming parameters.

5. Conclusions

In this paper, the problem of named entity recognition is formulated as one of the most popular problems of natural language processing. The search, recognition, and extraction of named entities find practical application in the tasks of text annotation and summarizing, named entities linked, creating chat-bots, sentiment analysis of information, etc. NER results are used for the automatic processing of data from social networks to predict people's intentions, the data extraction from medical electronic patient records, in the field of cybersecurity.

The generalized neural network algorithm for solving the NER problem consists of the following steps: setting the problem in terms of a given data domain, defining a set of named entities to search for, forming an incoming corpus of documents; pre-processing incoming data to be transferred into the neural network; creating a neural network of a certain architecture, training and testing it; analyzing the data obtained. The paper describes the BiLSTM-CRF architecture of a neural network and the peculiarities of its application for solving the NER problem.

To conduct experiments using the BiLSTM-CRF architecture, a corpus of text resumes consisting of 160 documents, manually labeled, is used as input data. For the search, 10 types of named entities are defined that are of value when analyzing resume data. The resume corpus was divided into three parts for training with validation and further testing of the neural network. The Precision, Recall, and F1-score metrics were used to evaluate the network performance. The average values of these metrics for data from 30 resumes of the test dataset were $P=79.06\%$, $R=71.51\%$, and $F1=75.09\%$. Their values show the high ability of the BiLSTM-CRF network to recognize specified named entities in a given data domain.

The described algorithm for solving the NER problem, the algorithm for pre-processing incoming data, and the presented approach to using the BiLSTM-CRF architecture are universal and can be applied to solving NER problems for various data domain and named entities. It should be noted that the quality of the neural network approach for NER solution largely depends on the input data pre-processing results, formed input documents corpus, learning dataset size, and requires a search for hyper parameters for each particular tasks. High quality tokenization and the use of additional dictionaries, like geographical location, can positively affect the rate of recognition and named entities classification. Manual labeling of the training dataset with named entities is time consuming and can be optimized by using automated tagging software.

The results of the experiments prove the existence of prospects for further development of the developed software application for solving applied problems of natural language recognition.

6. Acknowledgements

This work was supported by Dniprovsk State Technical University under the state science and research work “System analysis and computer modeling of technological processes and information technologies”.

7. References

- [1] S. Kamath, R. Wagh, Named entity recognition approaches and challenges, *International Journal of Advanced Research in Computer and Communication Engineering* 6 (2017) 259 – 262. doi: 10.17148/IJARCCCE.2017.6260.
- [2] I. Augenstein, L. Derczynski, K. Bontcheva, Generalisation in named entity recognition: a quantitative analysis, *Computer Speech & Language* 44 (2017) 61–83. doi: 10.1016/j.csl.2017.01.012.
- [3] E. Batbaatar, K. H. Ryu, Ontology-based healthcare named entity recognition from Twitter messages using a recurrent neural network approach, *International Journal of Environmental Research and Public Health* (2019) 1–19. doi:10.3390/ijerph16193628.
- [4] C. Colon-Ruiz, I. Segura-Bedmar, Protected health information recognition by BiLSTM-CRF, in: *Proceedings of the Iberian Language Evaluation Forum, IberLEF'19, Bilbao, Spain, 2019*, pp. 679–686.
- [5] A. Yu. Sirotina, N.V. Loukachevich, Named entities in cybersecurity: annotation and extraction, 2019. URL: http://www.dialog-21.ru/media/4669/upd-dialogue-2019_-student-session_sirotina_loukachevich.pdf
- [6] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, 2015. URL: <https://arxiv.org/abs/1508.01991>.
- [7] H. Kumar, A. Agarwal, R. Dasgupta, S. Joshi, A. Kumar, Dialogue act sequence labeling using hierarchical encoder with CRF, 2017. URL: <https://arxiv.org/abs/1709.04250>.

- [8] Z. Zhai, D.Q. Nguyen, K. Verspoor, Comparing CNN and LSTM character-level embeddings in BiLSTM-CRF models for chemical and disease named entity recognition, 2018. URL: https://www.researchgate.net/publication/327260366_Comparing_CNN_and_LSTM_character-level_embeddings_in_BiLSTM-CRF_models_for_chemical_and_disease_named_entity_recognition
- [9] A.V. Glazkova, Comparison of neural network models for classifying text fragments containing biographical information, *Software & System* 32 (2019) 263–267. doi:10.15827/0236-235X.126.263-267.
- [10] L.T. Anh, M.Y. Arkhipov, M.S. Burtsev, Application of a hybrid BiLSTM-CRF model to the task of Russian named entity recognition, in: *Proceedings of the Conference on Artificial Intelligence and Natural Language, AINL'18, St.Peterburg, Russia, 2018*, pp. 91–103. doi: 10.1007/978-3-319-71746-3_8.
- [11] M. Likasik, B. Dadachev, G.Simoes, K. Papineni, Text segmentation by cross segment attention, in: *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP'20, Stroudsburg, USA, 2020*, pp. 4707–4716.
- [12] O. Koshorek, A. Cohen, N. Mor, M. Rotman, J. Berant, Text segmentation as a supervised learning task, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, HLT'18, New Orleans, USA, 2018*, pp. 469–473. doi:10.18653/v1/N18-2075.
- [13] G. Guan, M. Zhu, New research on transfer learning model of named entity recognition, *Journal of Physics* 1267 (2019) 1–8. doi:10.1088/1742-6596/1267/1/012017.
- [14] S. D. Demianovych, A. A. Kramov, Method of noun phrase detection in Ukrainian texts, *Control systems and computers* (2019) 48–61. doi:10.15407/csc.2019.05.048.
- [15] A. M. Glybovets, Automated search of named entities in unmarked Ukrainian texts, *Artificial Intelligence* 2 (2017) 45–51.
- [16] Y. Li, T.Liu, D. Li, Q. Li, J. Shi, Character-based BiLSTM-CRF incorporating POS and dictionaries for Chinese opinion target extraction, in: *Proceedings of The 10th Asian Conference on Machine Learning, ACML'18, Beijing, China, 2018*, pp. 518–533.
- [17] O. O. Marchenko, Machine learning methods named entities recognition, *Problems in programming* 3 (2016) 150–157.
- [18] N. Patil, A. Patil, B. V. Pawar, Named entity recognition using conditional random fields, *Procedia Computer Science* 167 (2020) 1181–1188. doi: 10.1016/j.procs.2020.03.431.
- [19] S. Vijayarani, R. Janani, Text mining: open source tokenization tools – an analysis, *Advanced Computational Intelligence* 3 (2016) 37–47. doi: 10.5121/acii.2016.3104.
- [20] M. Naili, A. H. Chaibi, H. B. Chezala, Comparative study of word embedding methods in topic segmentation, *Procedia Computer Science* 112 (2017) 340–349. doi:10.1016/j.procs.2017.08.009.
- [21] Y. Xu, R. Goodacre, On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning, *Journal of Analysis and Testing* 2 (2018) 249–262. doi:10.1007/s41664-018-0068-2.
- [22] D. S. Batista: Named-entity evaluation metrics based on entity level, 2018. URL: http://www.davidsbatista.net/blog/2018/05/09/Named_Entity_Evaluation/