Named Entity Recognition Problem for Long Entities in English Texts

1st Oleksii Shatalov
Artificial Intelligence department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
0000-0002-7267-6718

2nd Nataliya Ryabova
Artificial Intelligence department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
0000-0002-3608-6163

***Abstract*—This paper is related to the problem of natural language processing (NLP), namely the named entity recognition (NER). This paper reveals the features of named entities recognition in English texts using deep learning (DL). The peculiarity of the study was the rather long length of the presented named entities: many of them could include a rather large number of words. The main problem was the amount of text that had to be recognized as a single entity. The results of the research are described here show the effectiveness of using deep neural network architectures for the task of recognizing long named entities in texts in English.**

Keywords—Natural Language Processing, Named Entity Recognition, Deep Learning, Convolutional Neural Network.

I. INTRODUCTION

The article is related to the problem of named entity recognition in English-language text – finding words or a sequence of words that describe a particular area of human activity, geographical objects, proper names, and so on. Thus, such a definition of words or sets of words of the same semantics significantly helps both in further analysis of the text by machine learning and deep learning models, determining before further processing only meaningful text fragments, the work of search engines, which can form relationships between different entities, building relationships in the Semantic Web, automation of business processes, and in human work, marking certain and significant semantically for people areas of the text. In addition, similar operations are carried out to classify and find fake news, create chat bots, create recommendation systems and so on.

Over the past few years, the number of various publications devoted to the neural network approach to natural language processing has been growing rapidly [1,2]. Of course, this is due to the new capabilities of deep neural networks, as well as the available powerful tools for working with them [3,4]. Deep neural networks demonstrate new capabilities, quality and speed improvements in solving such traditional NLP tasks as part-of-speech tagging, chunking, named entity recognition, semantic role labeling, classification and clusterization, text summarization [5]. Named entity recognition is one of the most popular and demanded tasks, but not always easy to implement. In fact, named entity recognition has become a separate area of NLP research with a pronounced practical focus and dependence on the specifics of subject areas, as well as tasks that need to be solved. There are many different approaches in this area. Often researchers focuses either on the specificity of the named entities themselves, which often depends on the subject area of application, or on the methods of machine and deep learning, which make it possible to increase the efficiency of the problem being solved [6]. However, recent studies are trying to combine several factors of influence on the solution of the task at once. Namely, they take into account the various specifics of named entities (the so-called flat, and, in contrast, nested ones), as well as the ever-increasing complexity of deep neural network architectures [7]. In this paper, we consider the possibly simpler, but at the same time quite effective approach for solving a specific applied problem of NER, using a deep neural network and taking into account the specifics of named entities.

Thus, the assignment was to develop a solution that could define sufficiently long named entities in terms of text, and to create a simple web application to interact with the developed solution.

II. RELATED WORK

Named entity recognition can also be considered as a task of identifying and/or extracting entities, which is an important part of the problem of extracting information and knowledge from various text sources. The result of solving this problem should be the classification of named entities into predefined categories, such as names of persons, names of organizations, geographical objects, etc. In recent years great progress in this area of research has been achieved through the use of various architectures of deep neural networks. At the same time, researchers increase the capabilities of their neural models through the creation of complex architectures, combining the properties of various deep neural networks [8]. Also the paradigm of deep active learning in which we actively select patterns to use during learning is used to support the dep learning approach [9]. The interesting results were demonstrated in [10], where named entity recognition problem is considered as graph-based parsing. Authors of this paper provided their model with a global view on the input via a biaffine model, based on deep biaffine attention for neural dependency parsing [11]. They considered two tipes of named entity recognition: flat NER models, which usually based on a sequence labeling approach and nested NER models, which containing refferences to other named entities. In fact authors considered named entity recognition as a structured prediction task and adopted a SoTA [11] dependency parsing approach for nested and flat NER. They used contextual embeddings as input to a multilayer BiLSTM. Also authors used a biaffine model to assign scores for all spans in a sentence. This approach to solving the NER problem proposed by these authors is quite effective, although quite general, and, in addition, requires large computing power.

III. A REVIEW OF EXISTING METHODS

Such a problem refers to the assignment of classification in the field of natural language processing. There are several

approaches for solving such a problem, which are listed below.

A. Classical methods

Most of them are based on rules or pre-built special dictionaries, which contain enumerated expected values that should be recognized as a named entity by assigning it to a particular class. Such dictionaries can be supplemented with custom values by developers who want to make their system more narrowly focused or simply extend its functionality.

B. Machine learning approaches

There are two main methods for machine learning to recognize named entities:

1) Multi-class classification: Named entities are represented as areas of text with a target class value to which they belong, and then machine learning algorithms are used to classify the text. The main disadvantage of this approach is the lack of perception of the context of use of words or groups of words in the named entity, since such algorithms cannot take into account the surrounding text of the target phrase.

2) Conditional Random Field (CRF): This approach allows you to analyze the sequence of data - the chain of target values of the classes of named entities, which helps the model to evaluate the previous inserted named entities and on the basis of these predict the class for the subsequent ones. Unfortunately, this model, due to the lack of above, as well as the complexity of preparing and tuning training, makes such an approach difficult to adapt to the real-world tasks of named entity recognition in industry.

C. Deep learning approaches

Currently the most popular for industrial scale use. The most popular and showing good accuracy are the models trained on marked-up data with architectures that allow to perceive the context of the surrounding text: recurrent neural networks (RNN), Long-Short Term Memory network (LSTM). Also, after the transfer of words in vector space, convolution operations are used to extract some features using Convolutional Neural Network (CNN) architecture. In addition, recently more and more solutions in the field of named entity recognition problems based on Transformer architecture are appearing. But at the moment, a significant problem is their training speed, as well as the amount of data needed for their acceptable training.

Thus, the most appropriate way to identify named entities in the text, suitable for our task, is the deep learning approach. In the further part of article we will describe the results of research related only to the use of this approach.

IV. Model Training Preparation

For the further research operations it is necessary to collect and search the dataset. Thus, we must define the subject area of the text on which the further work will be carried out as an example of the approach described in the article, the volume of the desired text, including the named entities themselves, the entities themselves that we are going to determine from the text, as well as the metrics that we will use to determine the quality of model learning. Among other things, there is a need for manual dataset labeling for the purity of the experiment, as well as the need to choose software for the task. We will discuss each point in the next section.

A. Subject area choosing

For a successful experiment, it is necessary to take non-traditional named entities that occupy a different amount of text each. In the research, the following candidates for the subject area were chosen:

- Technical specifications of different electronics products.
- Description of universities, their divisions, certain requirements at the time of the admission campaign.
- Job description.

During the comparative analysis according to different criteria, which included both the volume of existing text in the public domain for a given subject area and the complexity of obtaining this data and forming a certain structure from it for convenient further processing, the latter subject area was chosen, namely, the description of a job vacancy. We should also note that we only parsed the texts with free-text descriptions of various entities without a specific structure that would help the machine understand the location of various elements (named entities) based on this structure.

B. Determining the desired length of the text

To obtain the desired results, the text must be long enough to contain enough words to form long named entities.

After conducting a study on the average length of the texts of the selected subject area, as well as the specifics of writing different named entities there, a filter of 300 words per one text was formed.

C. Defining named entities

When working with the named entities of the presented subject area, several criteria for their selection were formed:

- Entities should not have a clear writing structure in the text.
- Entities must include useful information within the presented subject area.
- At least part of the entities should be presented in the text in an expanded format.
- The text volume of at least part of the named entities should include 4 words or more.
- The text of entities may include symbols in addition to those presented in the alphabet.

Taking into account the above-mentioned criteria for selecting entities for further research, as well as the subject area, the following entities were selected:

- Salary.
- Benefits.
- Education requirement with a description of the field or fields.
- Requirement for work experience with a description of the field or fields.

D. Metrics for assessing the quality of model learning

During the training, it will be necessary to use various indicators of the quality of training of the model. Thus, we can say that for further use of the presented solution it is important not only the fact of definition of entities, but also the erroneous definitions or non-determination of these entities. Therefore, the metric for the quality of the trained model was chosen as F1.

V. Dataset Selection and Labeling

A. Sources of text for the dataset

First of all, it is necessary to form a list of portals, from where it would be easy enough to take texts of job descriptions for further work with them. After analyzing the candidate sources of texts, 1200 job description texts for jobs from different areas of human activity from different sources of job offers were taken. Thus, it was decided that 1,000 texts would be used as a training part and the remaining 200 as a validation part.

B. Dataset labeling

An open source utility called doccano was used to label the dataset. One of the functions of this utility is to label text for named entity recognition tasks. Four entities were defined as described in the text above, and a detailed manual labeling of the dataset was performed.

1) Particulars of dataset labeling: During the datasetlabeling action, some controversial points were detected, which should be the same when labeling the entire sample and agreed upon before training the model. For example, in both the entity describing education and the entity describing work experience, not only the degree (for education) or the work experience measure with its value (for work experience) should be marked, but also the field in which the education or work experience was obtained. Among other things, benefits should be labeled one by one, so that only one benefit is mentioned in one highlighted entity. Also, when labeling a salary, pay attention to the fact that it can be called by different words and can be taken for different period of time with a definite numerical interval. Thus, it is necessary to take into account the above remarks during the markup of the texts and to smell each of them in sufficient detail.

2) Form for storing labeling results: After labeling we need to choose a specific format in which we are going to store the data before submitting it for training or validation. So, our team decided to store it in JSONL format, where each line is a separate JSON, and it contains fields such as "text", "entities" and other metadata. The "entities" field stores arrays of 3 elements, which are responsible, respectively, for the start position of the text symbol, the end position of the text symbol, and the name of the named entity to which the text between the two characters is related.

VI. Model Training

A. Choise of architecture

The architecture based on the convolutional neural network (CNN) was chosen to train the model. During the development of the solution of the task and the experiments it was the best architecture that gave the best results on a small volume of the input text in the training part of dataset.

To work with the text, the pre-trained word2vec vector space translation module will be used, which will allow to perform certain calculations with vector word representations, including determining the context of words or word sets, improving the process of classification of named entities and their location in the text itself, and this approach will allow to operate with quite a large volume of words.

The convolution operation will be used to find features in the text that will allow us to properly find the text fragment that contains the named entity we have chosen. Also, by constructing a matrix of vector word representations and performing a convolution operation on them, in a sense the model captures the context of the phrase in which the named entity may be present. It should also be noted that learning will be done with a sliding window on the text, thereby allowing the named entity to have a fairly large number of words within itself.

B. Equipment and duration of model training

The training was performed on the GPU computing device from NVIDIA GTX 1080 8GB, which gave a significant increase in the learning speed.

As mentioned above, training was performed on 1,000 texts, which were presented in the format obtained after labeling, in the form of JSONL file.

Several experiments were conducted and models were obtained to measure the accuracy metrics for identifying named entities on a validation sample of 200 texts. As the models were tested, part of the labels was changed, and the labeling rules, which were partially described in the last section, were supplemented in the learning process.

Eventually, a model that satisfied the accuracy requirements was obtained. The training duration was 6 hours and 52 minutes with no time spent on the intermediate run of training validation once every 25 epochs, and the number of epochs was 275.

C. Model training results

As a result, we obtained a model that satisfies the accuracy parameters and analyzes English-language texts with job descriptions for the presence in the text named entities that were presented above.

More detailed information about the model metrics can be seen below in Table 1.

TABLE I. Metrics values of the obtained model

Metric	Value
Accuracy	0.87
Precision	0.88
Recall	0.79
F1	0.83

Other examples of named entity recognition will be presented in the section below.

Note that because of the diversity of job topics and the fuzzy structure of job offer descriptions, the model is able to identify the represented named entities in various texts that may not be related to the sources from which the texts for the shadowing and validation samples were taken.

VII. Model Integration into the Web Application

For the further work of the software application and the convenience of user interaction with the trained model we will use a shell in the form of a web application, which will allow other applications to use the model through the API, as well as the GUI for interaction with the end user.

The development of the shell was carried out using the tools of the Python 3.8. In this way, the reliability and simplicity of the software creation process was guaranteed. To create the web application we used a micro-framework called Flask, which allows us to create simple controllers and generate view templates based on the program output.

We should also note that after the model defines a named entity in the input text, the output is approximately the same data format as the model training, namely, an array with 3 elements defining the initial and final character indices of the entity text, as well as its name. Therefore, within the development of applications, post-processing of the model output was also performed to more clearly present the results of the model to the end user.

An example of the model output can be found in Figure 1.

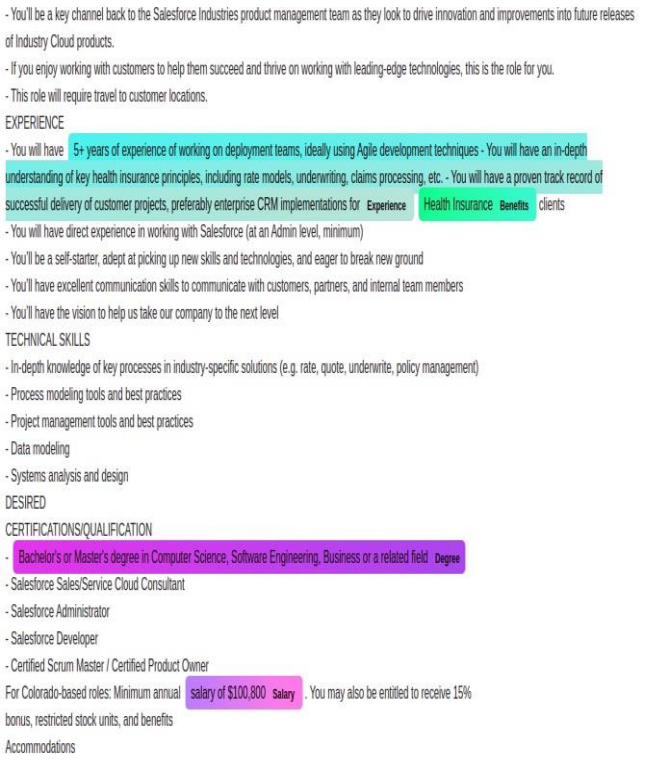

Fig. 1. Example of defining long named entities

VIII. Conclusions

The task to identify long (by the number of words) named entities in English-language texts was set, the subject area on which the results of the study were demonstrated, the named entities were selected, which allowed to fully demonstrate the idea of training the model to find long named entities.

A dataset was collected, which was subsequently processed and labeled with special labeling tool. The results of labeling were collected and formatted into a suitable form for the further use of the data by the model in training.

The result of the research was the model trained in a series of experiments, which shows good results, based on the metrics, the values of which are formed on the test sample. An off-the-shelf solution for classifying text with a complex description of one or another entity has been created. The model is able to perceive contextual information and form a sliding window to determine the location and class affiliation of the submitted part of the text.

The finished model has been integrated into a web application that provides a graphical interface for direct interaction with the end user, as well as an API for interaction with other software products that can use the presented functionality of the platform.

References

[1] Y. Goldberg, Neural Networks Methods for Natural Language Processing. Morgan&Claypool Publishing, 2017.

[2] L. Hobson, H. Cole, H. Hannes, Natural Language Processing in Action. Understanding, analyzing, and generating text with Python. Manning Publications Co, 2019.

[3] T. Ganegedara, Natural Language Processing with TensorFlow. Teach language to machines using Python's deep learning library, UK: Packt Publishing Ltd, 2018.

[4] D. Rao, B. McMahan, Natural Language Processing with PyTorch. Build Intelligent Language Applications Using Deep Learning, USA: O'Reilly Media, Inc., 2019.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, "Natural language processing (almost) from scratch," Journal of macine learning research, 12(Aug), pp. 2493–2537, 2011.

[6] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, "Neural architectures for named entity recognition," Proc. Of NAACL-HLT, California, San Diego, pp. 260–270, 2016.

[7] J. Li, A. Sun, J. Han, and C. Li, "A Survey on deep learning for named entity recognition," IEEE Transactions on Knowledge and Data Engineering, DOI: 10.1109/TKDE.2020.2981314.

[8] Jason P.C. Chiu, E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," Transactions of thr Association for Computational Linguistics, vol. 4, pp.357–370, 2016.

[9] Y. Shen, H. Yun, Z. Lipton, Y. Kronrod, A. Anandkumar, "Deep active learning for named entity recognition," Proc. 2nd Intern. Workshop on Representation Learning for NLP, Canada, Vancouver, pp. 252–256, 2017, DOI: 10.18653/v1/W17-2630.

[10] J. Yu, B. Bohnet, M. Poesio, "Named entity recognition as dependency parcing," Proc. of the 58th Annual Meeting of the Association for Computational Linguistics, https://aclweb.org/anthology/2020.acl-main.577.pdf, DOI: 10.18653/v1/2020acl-main.577.

[11] T. Dozat, C. Manning, "Deep beaffine attention for neural dependency parsing," Proc. of 5th Intern. Conference on Learning Representations (ICLR), 2017.