Named Entity Recognition in Information Security Domain for Russian

Sirotina Anastasiia
Lomonosov Moscow State University
(Russia)
overnastuhed@yandex.ru

Loukachevich Natalia
Lomonosov Moscow State University
(Russia)
louk_nat@mail.ru

Abstract

In this paper we discuss the named entity recognition task for Russian texts related to cybersecurity. First of all, we describe the problems that arise in course of labeling unstructured texts from information security domain. We introduce guidelines for human annotators, according to which a corpus has been marked up. Then, a CRF-based system and different neural architectures have been implemented and applied to the corpus. The named entity recognition systems have been evaluated and compared to determine the most efficient one.

1 Introduction

For a cybersecurity expert, it is vital to stay aware of all of the newly discovered vulnerabilities and exploits. Although various security databases, such as National Vulnerability Database (NVD) or MS Bulletins, contain detailed information on various cybersecurity problems, they do not usually provide any data on the latest discoveries. The most up-to-date descriptions of vulnerabilities are usually posted on specific websites and forums in form of **unstructured informal** texts. Therefore, a system extracting relevant cybersecurity information from unstructured publications could be of great use for a cybersecurity expert.

As there are a lot of NER systems for news documents, the first idea one comes up with is to apply such a system to the cybersecurity domain. Unfortunately, all such attempts prove to be unsuccessful for two reasons. Firstly, there is a great difference between general news documents and informal publications on cybersecurity: the latter contain a lot of non-vocabulary words (such as jargonisms, borrowings and domain-specific terms), include various grammar and spelling mistakes and use spoken syntax. What is more, changing the domain, where the extraction is conducted, usually means that some new named entity types should be accounted for (e.g. for cybersecurity domain these are names of programs and viruses). As none of the existing NER systems can be applied to the cybersecurity domain, our only option is to create a new system, which is trained on a newly labeled corpus, where all the domain-specific names and terms are taken into account.

This paper observes the named entity recognition (NER) task for unstructured Russian texts related to cybersecurity. Our first step is a thorough analysis of unstructured texts related to cybersecurity, elaboration of guidelines for human annotators and, finally, corpus labeling. At the second step, we implement several NER systems (one based on CRF-method and others based on artificial neural networks), apply them to the corpus, evaluate and compare of the results shown by the systems.

2 Related Work

The information extraction task in cybersecurity domain has been discussed in several works. However, the vast majority of the works consider information extraction only from structured or semi-structured English texts. For instance, (Bridges et al., 2013) and (Weerawardhana et al., 2014) use training corpora consisting of MS Bulletins and NVD vulnerability descriptions mainly. The training corpus presented in (Joshi et al., 2013) does contain unstructured blog posts, but those comprise less than 10% of the corpus.

NER systems elaborated in the works mentioned above are based on several different methods such as principle of Maximum Entropy in (Bridges et al., 2013), Conditional Random Fields (CRF) in (Weerawardhana et al., 2014: Joshi et al., 2013).

One of the latest works on the topic is (Gasmi et al., 2018). The authors use the corpus created by Bridges et al. (2013) as a dataset for two different NER systems. The corpus of over 850 000 tokens includes a large amount of NVD

Proceedings of Recent Advances in Natural Language Processing, pages 1114–1120,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_128

descriptions and also some MS Bulletins and Metasploit Framework's descriptions. The corpus was auto-labeled by a special algorithm. The labeling scheme is BIO. The two NER systems trained by Gasmi et al. (2018) are a CRF-model (CRFSuite implementation by Okazaki (2007)) and a neural network (NN) based model LSTM-CRF (as suggested by Lample et al. (2016)). The NN-based model combines bidirectional LSTM, word2vec models as a source of pre-trained word embeddings and CRFs as an output layer.

The only work that discusses the NER task for unstructured Russian text from cybersecurity domain is (Mazharov and Dobrov, 2018). The authors train their NER system on an early version of Sec_col collection, which was annotated with a CRF-classifier trained on news documents (Mozharova and Loukachevitch, 2016a).

3 Labeled Corpus Construction

The source of the text for our corpus is Sec_col collection. It consists of 2000 texts (posts and forum publications) from SecurityLab.ru website. These texts can be described as follows: they do not include any structured information; the writing style is informal; they contain quite a lot of lexical, spelling and grammar mistakes, many borrowings, words in foreign languages, words containing non-alphabetic characters and many instances of jargon. The average length of the texts in Sec_col collection is about 400 words.

The texts were manually marked up by four independent annotators, not all of which are cybersecurity experts. For the annotation purpose, the BRAT web based annotation tool was used.

There is no conventional set of labels for texts related to cybersecurity, but different authors usually consider quite similar classes of named entities to be relevant for the cybersecurity domain. For instance, both Bridges et al. (2013) and Joshi et al. (2013) annotate such types of named entities as software names, software versions, file names, vulnerability names. Taking into account the experience of the works mentioned above, we propose the following set of labels to annotate named entities: **Person** – people's names; **Loc** – locations; **Org** – names of organizations; **Hacker** – individual hackers' nicknames; **Hacker_Group** – names of hacker groups; **Program** – software products and parts of programs; **Device** – electronic gadgets; **Tech** – technologies; **Virus** – various malicious software; **Events** – for example, conferences.

At first, annotators had quite modest instructions on how various named entities must be annotated. Our assumption was that this would help to detect the trickiest contexts and regular mistakes that should be mentioned in the full-fledged guidelines for annotators.

In total 1124 publications have been marked up. Then those texts that do not contain any named entities that are relevant for cybersecurity were excluded. The final corpus contains 861 texts, which is more than 400 000 tokens.

In order to ascertain the degree of correctness and consistency of the annotation, we conducted a thorough analysis of the labeled texts. As the result, we found out that annotators tend to mark up similar contexts quite differently. For example, token *ICQ* was labelled as **Program** 18 times and received label **Tech** in 9 other cases.

Also some other instances of inconsistent markup were discovered:

- The number of named entities (one or two) in the contexts where an abbreviation is followed by the full name of the same entity or vice versa: *Software Defined Network (SDN)*; *MITM (Man-in-the-Middle)*;

- The number of named entities (one or two) in the contexts where a name or term in Russian is followed by the same name in English;

- Presence or absence of a named entity on a version of software or device when it is separated from the name by a punctuator or a conjunction: *Android 7.1, 7.1.1;*

- Inclusion or exclusion of paired punctuators (such as quotation marks or brackets) that surround a named entity: *сканер уязвимостей (nessus)* - (scaner ujazvimostej nessus) "vulnerability scanner nessus";

- Inclusion or exclusion of the second part of a compound noun with a hyphen, the first part of which is a named entity: *Android-устройств* – (android ustrojstv) "devices run on Android".

The vast variety of mistakes and instances of inconsistency reveal that there is a strong need for

Labels	Org	Loc	Person	Tech	Program	Device	Virus	Event	Hacker_group	Hacker
Amount of entities	3797	1553	1130	3280	3995	586	561	310	45	16

Table 1: Statistics of the annotated named entities.

guidelines for human annotators, whether they are cybersecurity experts or not.

3.1 Guidelines for Annotators

Within our study, comprehensive annotators' guidelines have been developed. It includes description of all the problematic and/or ambiguous contexts and indicates how named entities should be annotated in these contexts. Some passages from the guidelines are provided hereunder.

- If an abbreviation is followed by the full name of the same entity or vice versa, then the whole sequence is annotated as a single NE;
- If a name or a term in Russian is followed by the same name in English, then the whole sequence is annotated as a single NE;
- If a version of a software or a device is separated from its name by a punctuator or a conjunction, then it is annotated as a named entity only if it contains an alphabetic character, for example: *PowerPC G3, G4;*
- Paired punctuators (such as quotation marks or brackets) that surround a named entity are included in its annotation;
- The second part of a compound named entity with a hyphen is included in its annotation.

The guidelines also include detailed labels' description.

For each label we indicate the types of named entities that should receive this label. For example, label **Program** should be assigned to: operating systems (*iOS 9*); browsers (*Google Chrome*); programs that can be downloaded and installed (*Adblock*); websites (e.g. *SlideShare*, but not a link: *https://www.slideshare.net/);* files and processes, whose names are in format «name.extension» (*Autorun.exe*, *ipfilter.dat*) and some others. Likewise, label **Tech** should be assigned to: formats and filename extensions (when mentioned separately, e.g. *XML*); programming languages (*Javascript*); protocols (*pptp*), standards (*PCI DSS*) and some others.

For each label we also give the types of named entities that should not receive this label. For example, label **Program** should not be assigned to: links and directories, long instances of programmers' code, some abbreviations (*OS*). Likewise, label **Tech** should not be assigned to: abbreviations such as *BYOD, CYOD*, which are not technological, but business approaches.

The annotation of the corpus was corrected in accordance with the guidelines. To access how useful the introduction of the guidelines is, we could compare agreement between annotators before and after the guidelines were introduced. Unfortunately, at the current point annotator's agreement cannot be measured as there are no texts for which we would have several annotations from different annotators.

In Table 1 the statistics of the annotated named entities in the renewed corpus is presented.

The renewed corpus was used to train and test several NER systems based on the modern machine learning methods.

4 Labeling Scheme

In recent works on NER, several specific labelling schemes are usually used: IOB, BIO or IOBES. The general idea is to enrich standard labels with prefixes that indicate position of a token inside of a named entity: for example, whether it is the first word of a named entity.

It was shown that BIO-scheme is the one most efficient tagging schemes for NER system training: see (Mozharova and Loukachevitch, 2016a) for CRF-model and (Reimers and Gurevych, 2017) for neural networks. BIO-scheme suggests marking the first token of any named entity with a B- tag (beginning), every other token within the named entity should receive an I- tag (inside). Every token that is not a part of a named entity re-

List name	Examples of objects	Amount of objects
Device_name	*Macbook, Em Marine*	54
Device_type	*лэптоп* (laptop, “laptop”), *плеер* (player, “player”)	57
Hacker_descr	*хакер* (hacker, “hacker”), *онлайн-вор* (online-vor, “online-thief”)	31
Hacker_name	*UGNazi, AnonCoders*	39
Org_name	*ABBYY, ZYXEL*	306
Org_type	*холдинг* (holding, “corporate group”), *лаборатория* (laboratoriya, “laboratory”)	46
Program_name	*Amazon, Blackberry*	335
Program_type	*firewall, antispy*	167
Tech_name	*CD, SSH*	195
Tech_type	алгоритм (algoritm, „algorithm“), протокол (protokol, “protocol”)	19
Virus_name	*MITM, NonPetya*	42
Virus_type	Trojan, Malware	179

Table 2: Lists information.

ceives tag O (outside).

5 CRF-Model

CRF-model is a discriminative classifier, which can be used to predict sequences (Lafferty et al., 2001). A CRF-classifier uses information about the whole sequence of observable states (words) and information from previous unobservable states (labels). CRFs proved to be one of the most successful methods for NER.

To test a CRF-model, we use an open source implementation CRF++[1].

The following set of tokens’ features was used for the model training:

- **String features**: length of the token; initial letter’s case; presence of non-alphabetic characters; presence of a vowel, etc.
- **Token’s lemma**;
- **Part of speech (POS) of a token**;
- **Lexicon features**: whether a token is mentioned in a special vocabulary list (see below);
- **Cluster feature**: a number of the token’s cluster (see below);
- **Context features**: all the features mentioned above for the two preceding tokens and for the two following tokens;
- **Bigram feature**: previous token’s label.

To determine lemma and POS of each token, an open source morphological analyzer MyStem[2] was used. For tokens that do not have any lemma (for example, punctuation marks), lemma feature coincides with the token. For tokens that are absent in MyStem’s vocabulary, POS feature gets value ‘N\A’ or ‘PUNCT’, if a token is a punctuation mark.

We explain lexicon and cluster features in more details.

5.1 Lexicon Features

To improve NER system’s results we can use vocabularies that contain lists of objects of a certain type. By object we mean a word or a phrase. Words can also be treated as single-word phrases.

At our disposal there are 12 lists (see Table 2 for more detailed information). To create these lexicons, a large collection of texts from different sources on cybersecurity was used. Mutli-word terms and names were extracted from the collection. Then the extracted objects were manually classified into several categories.

Each token has 12 lexicon features. For a token T, a lexicon feature gets value ‘0’ if there is no word T or phrases that include T in the correspondent list. If token T is included in some phase in the list, then the correspondent lexicon feature gets value which equals the matched phrase’s length.

Let us consider an example. For token *Google* (as a part of a named entity *Google Chrome*), the feature that corresponds to Org_name list will get

[1] https://taku910.github.io/crfpp/

[2] https://yandex.ru/dev/mystem/

value ‘1’, while the feature that corresponds to Program_name list will get value ‘2’.

5.2 Cluster Feature

To form word clusters, we use an open source model ruwikiruscorpora_upos_skipgram_300_2_2019 (RusVectōrēs[3], (Kutuzov et al., 2016)). In total 300 clusters were formed. For each token the cluster feature get value that equals the number of the cluster, that contains the token. If there is no such a cluster, then the feature gets value ‘-1’.

6 Neural Networks

Nowadays the most successful NER systems are usually those that are based on neural networks.

Within our study, six different neural network architectures were implemented:

(A) BiDirectional LSTM: **BiLSTM**;

(B) BiDirectional LSTM with a CRF-classifier as an output layer: **BiLSTM-CRF**;

(C) BiDirectional LSTM with BiDirectional LSTM embeddings: **BiLSTM$_{\text{CHAR}}$-BiLSTM**;

(D) BiDirectional LSTM with BiDirectional LSTM embeddings and a CRF-classifier as an output layer: **BiLSTM$_{\text{CHAR}}$-BiLSTM-CRF**;

(E) BiDirectional LSTM with CNN embeddings: **CNN$_{\text{CHAR}}$-BiLSTM**;

(F) BiDirectional LSTM with CNN embeddings and a CRF-classifier as an output layer: **CNN$_{\text{CHAR}}$-BiLSTM-CRF**;

The core layer in all the architectures is Bidirectional Long-Short Term Memory (BiLSTM) NN (Graves et al., 2013), (Huang et al., 2015). BiLSTM is capable of learning long-term dependencies and considers both left and right context of every token.

All the architectures use pre-trained word embeddings created by model araneum_none_fasttextskipgram_300_5_2018 (RusVectōrēs, (Kutuzov et al., 2016)). Fasttext models are able to build embeddings for non-vocabulary words (e.g. jargonisms or borrowings), which is vital for a big corpus like ours.

Models (B), (D) and (F) use a CRF-classifier as an output layer (Huang et al., 2015), (Lample et al., 2016), (Ma et al., 2016). Therefore, these models are capable of learning standard constraints of the markup such as that a token with I-label must always follow a token with B-label of the same class.

Models (C)-(F) also have special layers that build character embeddings (Lample et al., 2016), (Ma et al., 2016), which is said to improve the results shown by NER systems (Reimers and Gurevych, 2017), (Zhai et al., 2018). While models (C)-(D) use BiLSTM-layer to build character embeddings, models (E)-(F) use CNN-layer for the same purpose. Reimers and Gurevych (2017) and Zhai et al. (2018) have shown that both layers provide the same improvement of a NER system, but CNN-layer is characterized by higher computational efficiency.

7 Evaluation

To evaluate all the NER systems, standard metrics such as Precision, Recall and F-score were used for each named entity type (label).We also compute macro and micro metrics for each system. The evaluation method is as follows: a named entity considered to be correctly identified by a NER system (true positive decision) only when both the type and the boundaries are correctly defined. This method is called exact (full) matching method.

To calculate the metrics the 3:1 cross validation technique was used.

Table 3 presents Precision for all the systems, while Table 4 and Table 5 present Recall and F-measure, respectively. The letters in the head of the tables stand for NN-based models (as it was introduced in Section 6). We also use following notations: **P** for **Person** ; **L** for **Loc**; **O** for **Org**; **H** for **Hacker**; **Hg** for **Hacker_Group**; **Pr** for **Program**; **D** for **Device**; **T** for **Tech**; **V** for **Virus**; **E** for **Events**; **Ma** for macro measures; **Mi** for micro measures.

As we can see, the CRF-model outperforms all the NN-based models. A possible explanation could be the fact that only CRF-model uses lexicon features.

As far as NN-based models are concerned, BiLSTM$_{\text{CHAR}}$-BiLSTM-CRF proves to be the most successful one, judging from micro and macro metrics. CNN$_{\text{CHAR}}$-BiLSTM-CRF shows quite similar results and, as it was expected, it

[3] http://rusvectores.org/ru/about/

	CRF	**(A)**	**(B)**	**(C)**	**(D)**	**(F)**	**(F)**
O	**85.9**	68.7	73.0	75.3	78.1	78.3	76.4
L	**96.7**	90.2	88.1	92.7	92.9	95.5	94.6
P	85.4	28.9	61.2	79.1	**85.7**	72.8	79.2
H	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Hg	**87.5**	0.0	0.0	0.0	0.0	0.0	0.0
Pr	82.1	56.6	65.1	77.6	**85.8**	71.4	78.5
D	**65.4**	0.0	0.0	0.0	11.1	18.8	11.9
T	71.3	63.0	67.2	71.8	**77.4**	70.2	76.6
V	**68.5**	0.0	0.0	0.0	37.5	3.0	23.8
E	67.8	0.0	0.0	0.0	71.4	0.0	37.6
Ma	**71.0**	30.7	35.5	39.7	54.0	41.0	47.9
Mi	**82.2**	63.1	70.0	76.9	79.7	74.5	78.4

Table 3: Precision.

	CRF	**(A)**	**(B)**	**(C)**	**(D)**	**(F)**	**(F)**
O	65.5	30.3	38.3	62.1	**69.1**	48.6	67.5
L	81.9	39.4	53.5	70.0	**82.3**	52.5	73.5
P	**57.8**	8.9	30.0	46.9	54.7	35.0	49.1
H	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Hg	**14.0**	0.0	0.0	0.0	0.0	0.0	0.0
Pr	**61.2**	29.0	40.4	51.3	60.0	57.1	58.2
D	**21.9**	0.0	0.0	0.0	0.8	2.5	0.8
T	53.6	4.1	16.8	**55.5**	41.9	48.0	53.7
V	**28.3**	0.0	0.0	0.0	5.1	0.4	3.8
E	**27.2**	0.0	0.0	0.0	5.9	0.0	7.2
Ma	**41.1**	11.2	17.9	28.6	32.0	24.4	31.4
Mi	**59.0**	21.6	31.3	51.7	55.3	45.4	55.2

Table 4: Recall.

	CRF	**(A)**	**(B)**	**(C)**	**(D)**	**(F)**	**(F)**
O	**74.3**	42.0	50.2	68.1	73.3	59.9	71.6
L	**88.6**	54.8	66.6	79.8	87.3	67.6	82.7
P	**68.9**	13.5	40.3	58.9	66.8	47.2	60.6
H	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Hg	**24.0**	0.0	0.0	0.0	0.0	0.0	0.0
Pr	70.0	38.4	49.9	61.8	**70.6**	63.4	66.6
D	**32.5**	0.0	0.0	0.0	1.5	4.3	1.3
T	61.1	13.3	26.9	62.6	54.4	57.0	**63.1**
V	**39.6**	0.0	0.0	0.0	9.0	0.7	6.6
E	**38.5**	0.0	0.0	0.0	10.9	0.0	12.0
Ma	**49.7**	16.2	23.4	33.1	37.4	30.0	36.4
Mi	**68.7**	32.2	43.3	61.8	65.3	56.4	64.7

Table 5: F-score.

outperforms BiLSTM$_{CHAR}$-BiLSTM-CRF model in training time.

As for the results shown for different classes of named entities (labels), we could also suggest several explanations. The poor quality for **Hacker** and **Hacker_group** could be explained by the small amount of named entities of these classes in the corpus (16 and 45 respectively). The reason for the low scores for **Virus** is possibly semantic heterogeneity of the class, which, according to our latest guidelines, comprises both malicious software and various technologies that hackers use (e.g. *DDos*). Therefore, **Virus** class is also semantically similar to **Tech** and **Program** classes, which could also influence the scores. As for **Event** class, it is also semantically heterogenic, as the label is assigned to both human arranged events (e.g. seminars and conferences) and historical and cultural events (e.g. holidays and wars).

Unfortunately, the results performed by our models cannot be compared to the results in (Mazharov and Dobrov, 2018) for two reasons.

Firstly, although the same text collection Seq_col was used in both studies, the markup of the collection differs significantly. Only about 300 texts from dataset in (Mazharov and Dobrov, 2018) were annotated manually and contained labeled named entities that are relevant for cybersecurity. Other 1700 texts in the dataset had poor quality automatic annotation, provided by a CRF-classifier trained on news documents. In our study the dataset contains 861 texts that were manually marked up in accordance with the annotators' guidelines.

Secondly, as it was mentioned above, we used full matching method of evaluation, whereas the method used in (Mazharov and Dobrov, 2018) is incomplete matching.

8 Conclusion and Future Work

In this paper we discuss the NER task for unstructured Russian texts concerning cybersecurity problems. Our first step is creating a labeled corpus of such texts. In order to ensure correctness and consistency of the markup, we elaborate detailed annotators' guidelines, which include description of every label and numerous examples of various tricky contexts. Within our study, the first consistently labeled corpus of unstructured Russian texts on cybersecurity was created. The corpus can now be used either as a dataset for NER systems or to conduct linguistic analysis of the text in question.

Our guidelines can be used to create new labeled corpora of texts on cybersecurity or to annotate the rest of the texts in Sec_col to increase our corpus.

We applied several NER systems based on CRF method and on NN to our corpus. The most successful is the CRF-model. Our hypothesis is that the CRF-model outperforms all the other models because it is the only one that uses lexicon features. Among NN-based models, BiLSTM$_{CHAR}$-BiLSTM-CRF shows the best results.

As for the future work, the usefulness of our annotators' guidelines should be captured by comparing agreement between annotators before and after the guidelines were introduced. What is more, the set of features for the CRF-model could be widen by adding such features as text statistics or text collection statistics (Mozharova and Loukachevitch, 2016b). Furthermore, some additional features such as lexicon features and casing features could also be used to improve the performance of NN-based NER systems.

Acknowledgments

This work has been supported by the Russian Foundation for Basic Research (project 16-29-09606).

References

Robert A. Bridges, Corinne L. Jones, Michael D. Iannacone, Kelly M. Testa, & John R. Goodall, (2013). Automatic labeling for entity extraction in cyber security. arXiv preprint arXiv:1308.4941.

Houssem Gasmi, Abdelaziz Bouras, & Jannik Laval, (2018). LSTM Recurrent Neural Networks for Cybersecurity Named Entity Recognition. ICSEA 2018, 11.

Alan Graves, Abdel-rahman Mohamed, & Geoffrey Hinton, (2013, May). Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 6645-6649). IEEE. https://doi.org/10.1109/ICASSP.2013.6638947

Zhiheng Huang, Wei Xu, & Kai Yu, (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

Arnav Joshi, Ravendar Lal, Tim Finin, & Anupam Joshi, (2013, September). Extracting cybersecurity related linked data from text. In 2013 IEEE Seventh International Conference on Semantic Computing (pp. 252-259). IEEE. https://doi.org/10.1109/ICSC.2013.50

Andrej Kutuzov, & Elizaveta Kuzmenko, (2016, April). WebVectors: a toolkit for building web interfaces for vector semantic models. In International Conference on Analysis of Images, Social Networks and Texts (pp. 155-161). Springer, Cham. https://doi.org/10.1007/978-3-319-52920-2_15

John Lafferty, Andrew McCallum, & Fernando Pereira, (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, & Chris Dyer, (2016). Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360.

Xuezhe Ma, & Eduard Hovy, (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. arXiv preprint arXiv:1603.01354.

Ivan Mazharov & Boris Dobrov (2018). Named Entity Recognition for Information Security Domain.

Valeriia Mozharova, & Natalia Loukachevitch, (2016, April). Combining knowledge and CRF-based approach to named entity recognition in Russian. In International Conference on Analysis of Images, Social Networks and Texts (pp. 185-195). Springer, Cham. https://doi.org/10.1007/978-3-319-52920-2_18

Valeriia Mozharova, & Natalia Loukachevitch. (2016, August). Two-stage approach in Russian named entity recognition. In 2016 International FRUCT Conference on Intelligence, Social Media and Web (ISMW FRUCT) (pp. 1-6). IEEE. https://doi.org/10.1109/FRUCT.2016.7584769

Naoaki Okazaki, (2007). Crfsuite: a fast implementation of conditional random fields (crfs).

Nils Reimers, & Irina Gurevych, (2017). Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. arXiv preprint arXiv:1707.06799.

Sachini Weerawardhana, Subhojeet Mukherjee, Indrajit Ray, & Adele Howe, (2014, November). Automated extraction of vulnerability information for home computer security. In International Symposium on Foundations and Practice of Security (pp. 356-366). Springer, Cham. https://doi.org/10.1007/978-3-319-17040-4_24

Zenan Zhai, Dat Nguyen, & Karin Verspoor, (2018). Comparing CNN and LSTM character-level embeddings in BiLSTM-CRF models for chemical and disease named entity recognition. arXiv preprint arXiv:1808.08450.