Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2022"

Moscow, June 15–18, 2022

Sodner for Russian nested named entity recognition

Kirill Abrosimov
Seldon
Nizhny Novgorod, Russia
abrosimov.k@myseldon.com

Arina Mosyagina
Seldon
Nizhny Novgorod, Russia
mosyagina.a@myseldon.com

Abstract

The article describes the solution for Russian nested named entity recognition that we presented in the RuNNE competition. The solution is based on the Sodner model that predicts named entities in a text as a graph. During the competition we improved the training dataset and annotated the additional corpus that contains entities of the few-shot classes. After several experiments with different model parameters high macro F1 and few-shot F1 scores were obtained – 74.08 and 64.41 respectively.

Keywords: named entity recognition; nested named entity recognition; NER; few-shot NER; Sodner
DOI: 10.28995/2075-7182-2022-21-1-7

Sodner для извлечения вложенных именованных сущностей на русском языке

Абросимов К. И.
Seldon
Нижний Новгород, Россия
abrosimov.k@myseldon.com

Мосягина А. Г.
Seldon
Нижний Новгород, Россия
mosyagina.a@myseldon.com

Аннотация

В статье описано решение задачи извлечения вложенных именованных сущностей для русского языка, представленное в рамках соревнования RuNNE. Для решения была использована модель Sodner, представляющая именованные сущности в тексте в виде графа. В ходе соревнования нами был доработан обучающий датасет и размечен дополнительный корпус текстов, содержащих сущности few-shot классов. В результате нескольких экспериментов с различными параметрами модели были получены высокие значения macro F1 и few-shot F1 – 74.08 и 64.41 соответственно.

Ключевые слова: извлечение именованных сущностей; извлечение вложенных именованных сущностей; NER; few-shot NER; Sodner

1 Introduction

Named entity recognition (NER) is a task that aims at identifying objects of certain classes (e.g. Person, Organization, Location etc.) in a given unstructured text. Practical applications of this task include news and social media analytics, content recommendation systems and extracting data from medical texts. Different use cases require different classes of entities. Mostly, named entities are proper nouns, but they can also be common nouns or numbers (phone numbers, quantities, dates).

One of the NER's subtasks is nested named entity recognition (NNE). The main difference of NNE is that a named entity in this case can be a part of a larger one. For instance, the text span *"Московский государственный университет имени М.В.Ломоносова"* contains two entities: the whole span is an organization and the *"М.В.Ломоносова"* part is a person's name. The contrary to NNE is flat named entities, within which other objects cannot be identified by NER algorithms. Most of the existing solutions for the NER task only deal with flat named entities, while in real-world text data nested named

entities are more frequent. Many current methods, which are used for flat named entity recognition, treat NER as a sequence labelling task. However, sequence-based methods can also be used for NNE, as well as the methods based on hypergraphs. Nested named entity recognition in Russian is a novel task, for which the NEREL dataset was presented by the RuNNE shared task organizers [17].

NER models usually require large in-domain labeled datasets, although in many cases available amounts of data are very limited and annotating a training dataset might be too resource-consuming. For this reason, the few-shot NER was introduced. In a few-shot surrounding, a NER model is trained on just a few labeled examples of a certain class. The NEREL dataset includes 3 few-shot classes of named entities [4].

In the given article our team presents the solution for the RuNNE competition. The article is structured as follows: in section 2 the overview of the existing NNE and few-shot NER algorithms is provided; section 3 is devoted to the training data; section 4 describes our solution for the task; in section 5 the obtained results and analysis are presented, and the conclusion is made in section 6.

2 Related work

Most of the current solutions for the nested named recognition task use sequence-based methods. For instance, in [8] the authors propose a sequence-to-set model that predicts a set of named entities in a text at once, which is its main difference from typical sequence-to-sequence NER models that identify one entity at a time and are more suitable for flat named entities. Sequence-based methods can also be used while stacking model layers. In [3] the NNE task is solved by stacking flat NER layers of a neural network. Each flat NER layer comprises a BiLSTM layer and a CRF layer. In [10] a partly-layered network architecture is used for identifying nested and overlapping named entities.

Other methods include detecting boundaries of the entities in a text. A case in point is the HIT model that was proposed in [12]. This model leverages entity boundaries and connections between its internal tokens without considering their order. In [6] the authors present a two-stage NER identifier that was inspired by the task of object detection in computer vision. The process consists of locating named entities in the text by accurately defining their boundaries and labeling the located spans later. The authors of [7] use a span-based method that is enhanced with boundary detection to predict words that are boundaries of entities. Hypergraph-based methods are another type of methods for nested named entity recognition. In this case, segmental hypergraph representation is used for entity modeling [11].

Three methods of solving the few-shot NER task are described in [2]: prototype-based methods, self-training and noisy supervised pre-training. The authors of the article use a pre-trained RoBERTa model as a base model for the experiments. Prototype-based method, which they introduce, is similar to the previous methods in terms of using nearest neighbors for choosing an entity label. The difference of this method is that only prototypes, not separate tokens, are considered during comparison. The StructShot method proposed in [13] is based on structured neighbor learning. During training the method uses a supervised NER model, which is trained on a source domain, for feature extraction. The architectures, which the authors use, are BiLSTM and BERT-based model. For prediction, the given method uses a nearest neighbor classifier and a Viterbi decoder for modeling label dependencies. The MUCO model described in [9] utilizes O-class entities to induce new undefined classes of entities. Such classes are later jointly classified with predefined classes. For identification of undefined classes, a prototypical network is trained on predefined classes, and the O-class entities, which typically cluster during training, are classified according to clusters.

In [5] NER is treated as a language modeling task, which is another method for few-shot NER. In this case, a pretrained language model is fine-tuned to predict a class label word in an entity position in a given text. One of the current state-of-the-art methods for NER in a few-shot setting is CONTaiNER. This method uses contrastive learning to optimize inter-token distribution distance [1].

3 Data

Training data for the model was provided by the RuNNE competition organizers. The NEREL dataset comprises more than 900 Russian Wikinews articles and can be used for training not only named entity recognition, but also relation extraction models. NEREL includes 29 types of named entities in total, and training model to recognize 3 of them required a specific few-shot method [4].

During the analysis of the provided dataset some inconsistencies were identified, so the following corrections were made:

- all prepositions were included inside DATE objects' borders
- if a DATE object was followed by the word *"года"* (i.e. *"17 июня 2018 года"*), this word was also added inside the object's borders
- street suffixes such as *"улица"*, *"проспект"* and *"площадь"* were added into the FACILITY objects
- contexts "гражданин/гражданка/граждане + COUNTRY" were marked up as NATIONALITY with keeping the COUNTRY object as an internal entity
- occasional errors such as missed or intersecting entities were corrected
- punctuation marks were excluded from the entities

Some of the corrections were performed automatically and some manually.

One of the most frequent inaccuracies in the NEREL dataset was different number of nested entities within an object of the same class. A case in point is PROFESSION class: if an entity of this class had an embedded geopolitical entity, two different ways of annotation, which did not depend on the context, were observed. According to one of them, only the geopolitical object should be annotated as a nested entity: *[президент [США]$_{COUNTRY}$]$_{PROFESSION}$*. However, in the approximately same number of cases the same text span in the similar context can be annotated differently, namely the separate profession name is also an internal entity: *[[президент]$_{PROFESSION}$ [США]$_{COUNTRY}$]$_{PROFESSION}$*. To uniformize the training corpus, all of the embedded single profession names were annotated as PROFESSION entities. This principle of uniformity was also applied to the LAW class, e.g. for the entities such as *[[УК]$_{LAW}$ [РФ]$_{COUNTRY}$]$_{LAW}$*. During the experiments, our model was first trained on the original dataset and then on the cleared one. The results were compared, and the F1 score of the model trained on the cleared corpus increased by 1% compared to the original NEREL.

For the training model to identify objects of the few-shot classes (PENALTY, WORK_OF_ART and DISEASE) an additional training corpus was used. For this corpus 480 news texts were collected and manually annotated. The source of the texts is the company's news database, and certain SQL masks were used as a search criterion. During the annotation process only the fragments (from 1 to 3 sentences) that included at least one entity of any few-shot class were used.

4 Experimental setup

To solve this problem, the Sodner [14] model (A Span-Based Model for Joint Overlapped and Discontinuous Named Entity Recognition) was used. It was developed by researchers from Wuhan University and Tianjin University.

4.1 Sodner model

The model predicts named entities in the form of a graph, the vertices of which represent the entities, while the edges represent the relationships between entities (Entity Fragment Relation Graph). For the purposes of the competition the relations of nesting entities and the relations of discontinuity of the entity were used. The Sodner model consists of the following parts: word representation, graph convolutional network, span representation and joint decoding. A pre-trained Russian language BERT [15] by DeepPavlov was used for word representation, however Sodner can also work with such models as ELMO and Word2Vec with the embeddings being additionally transmitted to the Bi-LSTM network. Attention-guided graph convolutional network (AGGCN) [16] shows good results in processing syntactic dependency in text for entity extraction. This model consists of three parts: the multi-head attention mechanism is applied to the adjacency matrix to extract various patterns in the form of a matrix of weights of the connections of the vertices of the graphs. The results of each of the heads are processed by a Densely Connected Layer. The resulting matrices are fed to the linear combination to combine data into hidden representations using a fully connected network. As a result, the relevant information is extracted from the text parsing. Afterwards, a span enumeration is performed to consider each fragment of the text. The final decision is made in two stages: the first stage is the extraction of all named entities; the second stage is the pairwise classification of extracted entities to detect the dependencies between

them. Fully connected neural networks are used for each of the stages. As a result, the model uses text as input, and a graph representation of the text parsing in the format of a symmetric adjacency matrix. The model's output is a graph dependency of named entities. The natasha library was used for syntactic analysis during the competition.

Let's consider the following example from the test sample: *"Говорят, что бывший президент Чувашии развел в министерстве кумовство", пишут журналисты интернет-издания.* Figure 1 shows the adjacency matrix of parsing, which will be processed by AGGCN.

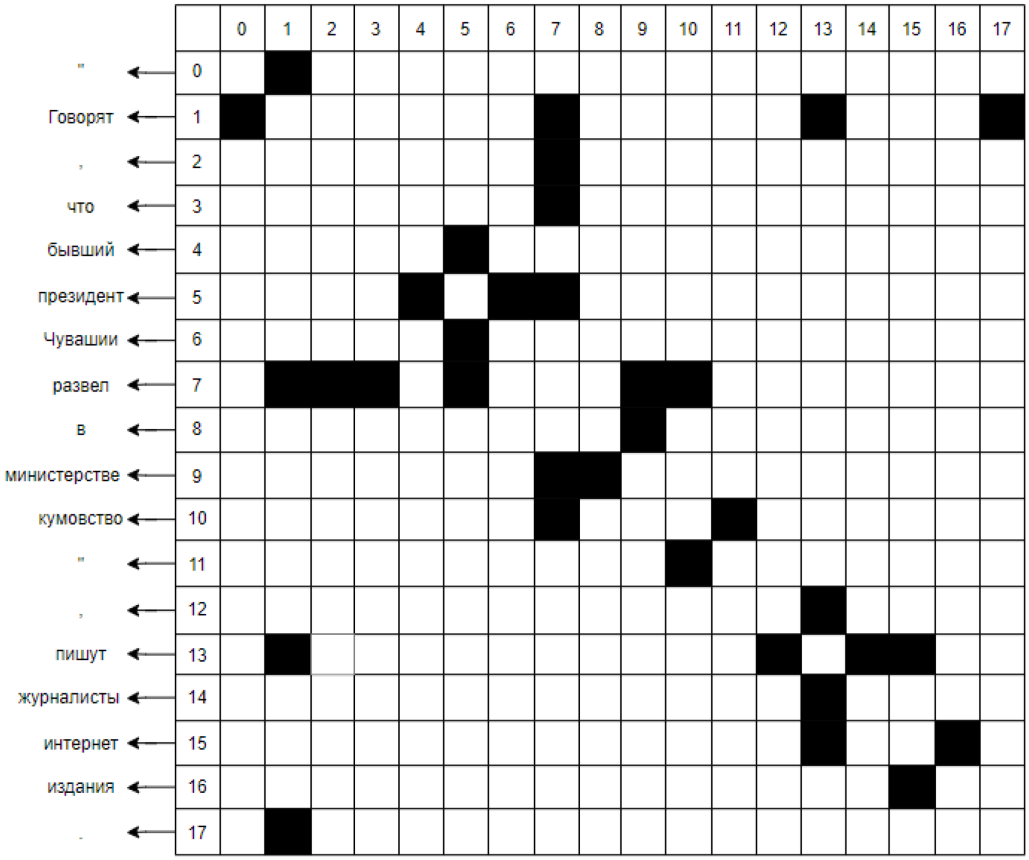

Figure 1: Adjacency matrix of sentence parsing

With the help of a pre-trained BERT, we obtain a word representation. After processing the received data, the Sodner model returns a graph shown in Figure 2.

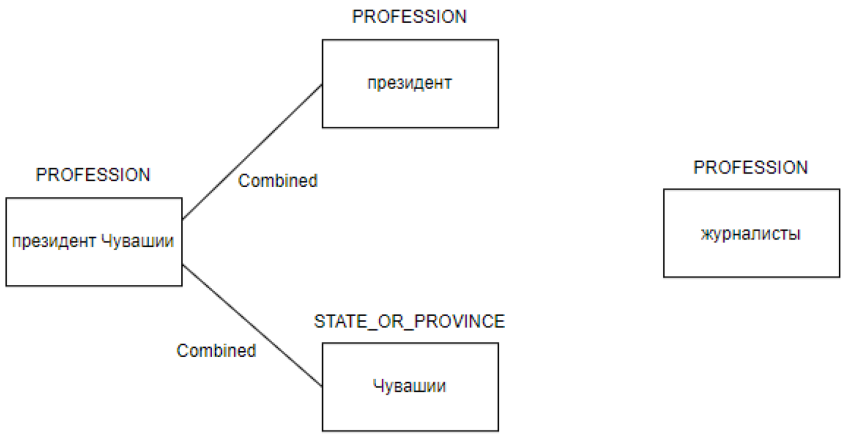

Figure 2: NER predicted by Sodner

4.2 Conducting experiments

A large number of experiments were conducted with various model configurations and datasets. The training of the Sodner model on GPU takes 10-12 hours on average. The main metrics in the competition were f1-scores: macro f1-score for regular NER, excluding few-shot classes, and macro f1-score for few-shot entities. All the metrics presented in the tables below are obtained from the final test sample.

5 Results and discussion

Next, we introduced a simple Sodner, which consists of 2 GCN layers with dropout 0.4, a vector representation of depth features dimension of 20, with 1 MLP layer of 150 neurons with dropout 0.4 and also uses an additional LSTM layer to represent words after the pre-trained BERT. First, we learned how much correcting errors in the layout of the dataset improves the prediction result of a simple Sodner and also how much the recognition of few-shot entities will improve thanks to additional marked-up data.

Data	Mention F1	Mention Recall	Mention Precision	Macro F1	Macro F1 few-shot
Original	81.88	80.37	83.44	72.17	54.56
Cleared	**82.53**	**81.29**	**83.81**	**73.13**	56.94
Augmented	81.57	79.50	83.76	72.39	**60.87**

Table 1: A simple Sodner trained on various data

Results in Table 1 allow us to conclude that the corrected version of the marked-up data has improved the metrics of the predictive model. However, there is always a possibility of incorrectly labeled named entities being present in the test sample. For example, the first result with extracted entities had an indexing error. For the phrase "*Российская Федерация в прошлом.*" indices [0, 20] were extracted instead of the true ones [0, 19]. However, even with this error, the result did not equal 0, which means there are errors in the markup of the test data. Additional data allowed the model to better extract few-shot entities. The next step consisted of configuring the model to obtain more usable results. There are a lot of settings in the model, some of which can help improve the metrics. The table below shows the main settings: the AGGCN block is represented by the number of layers, dropout, depth features dimension (dfd); the MLP block is the number of layers with the LSTM layer being used for additional information about words. The rest of the parameters are the recommended values specified in the article by the model developers. All experiments were carried out using an augmented cleared dataset.

GCN					Metrics				
layers	drop	dfd	MLP	LSTM	Mention F1	Mention Recall	Mention Precision	Macro F1	Macro F1 few-shot
2	0.4	20	1	+	81.57	79.50	83.76	72.39	60.87
1	0.2	64	2	+	82.23	80.63	83.89	73.78	59.35
2	0.2	64	2	–	**83.01**	81.14	**84.97**	**74.12**	61.85
3	0.2	64	3	–	82.77	81.62	83.95	72.56	**64.41**
3*	0.2*	64*	3*	–*	82.44*	**81.69***	83.20*	72.45*	**64.41***
4	0.2	64	4	–	82.79	81.52	84.10	72.30	64.29

Table 2: Sodner settings

Table 2 shows various model settings and the resulting metrics. The asterisk indicates the same model, but with additional named entities that were obtained using the NER model pre-trained on the natasha library. The models showed the biggest improvement when the vector representation of the GCN text was increased (from 20 to 64), as well as after disabling the LSTM layer. The resulting solution was an ensemble of the two best models that was created based on the following principle: all few-shot entities were taken from one model, and the remaining classes of named entities were taken from the other. In

total, the following metrics were obtained: macro f1 – 74.08, macro f1 few-shot – 64.41. Figure 3 shows the scheme that allowed us to obtain this result.

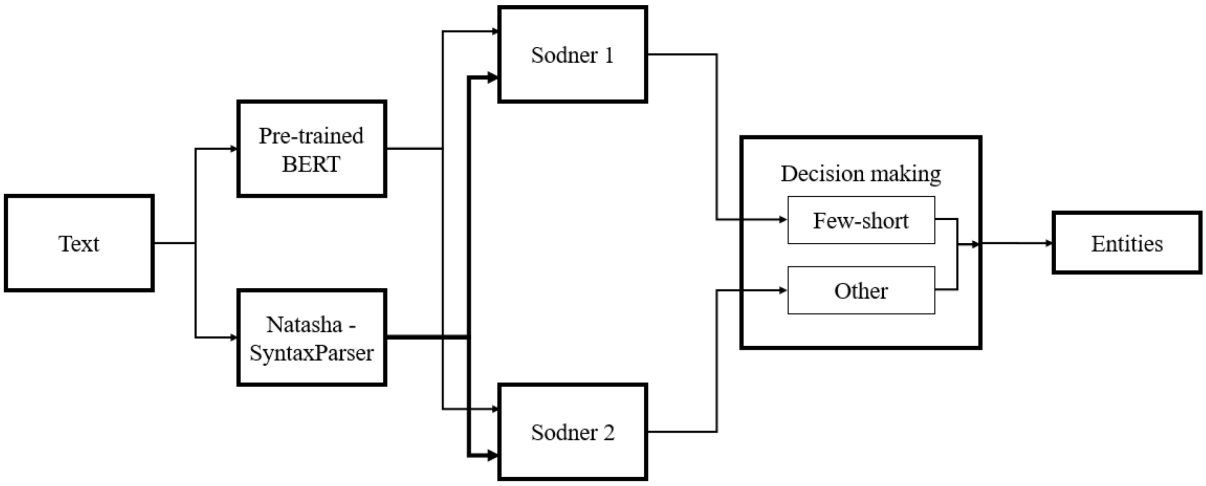

Figure 3: Our model that showed the best result

The presented solutions, despite very good results exceeding the baseline, have a number of difficulties. One of such difficulties is obtaining a parsing matrix, which means the quality of the nested entity extraction model itself depends on the quality of the resulting adjacency matrix. It is also worth noting that the model is not aimed at solving tasks with few-short classes. Therefore, a small collection of examples of a certain class results in the model being undertrained on these classes and the extracted few-shot classes are labeled incorrectly. As part of further research on the Sodner model, it can be noted that experiments were conducted on the coefficients of importance of the problem – the extraction of the entity or extraction of the classes of connections between these entities, which are used in the loss function during the model training. One of the advantages of the model is the ability to extract not only nested entities, but also discontinuous ones. Such entities were present in the NEREL dataset, but they were not enough for high-quality extraction of such relations. Not all experiments with settings are presented in the final table, only the most significant ones. Not all the parameters were experimented with, which could also help improve the results.

6 Conclusion

As a result of participating in the RuNNE competition, our team was able to achieve excellent results, surpass the baseline model by editing and correcting inaccuracies in the NEREL dataset, adding additional marked-up texts with few-shot entities, as well as using the Sodner model with various parameters. We were able to get macro f1 – 74.08 and macro f1 few-shot – 64.41, which is 6.5% and 19.75% higher than the baseline metrics (macro f1 – 67.44, macro f1 few-shot – 44.66).

References

[1] Das Sarkar Snigdha Sarathi, Katiyar Arzoo. CONTaiNER: Few-Shot Named Entity Recognition via Contrastive Learning — 2021. — Vol. arXiv:2109.07589. — version 1. Access mode: https://arxiv.org/abs/2109.07589v1.

[2] Huang Jiaxin, Li Chunyuan et al. Few-Shot Named Entity Recognition: A Comprehensive Study — 2020. — Vol. arXiv:2012.14978. — version 1. Access mode: https://arxiv.org/abs/2012.14978v1.

[3] Ju Meizhi, Miwa Makoto et al. A Neural Layered Model for Nested Named Entity Recognition // Proceedings of NAACL-HLT 2018. — New Orleans, Louisiana, USA, 2018. — P. 1446–1459.

[4] Loukachevitch Natalia, Artemova Ekaterina et al. NEREL: A Russian Dataset with Nested Named Entities, Relations and Events — 2021. — Vol. arXiv:2108.13112. — version 2. Access mode: https://arxiv.org/abs/2108.13112v2.

[5] Ma Ruotian, Zhou Xin et al. Template-free Prompt Tuning for Few-shot NER — 2021. — Vol. arXiv:2109.13532. — version 1. Access mode: https://arxiv.org/abs/2109.13532v1.

[6] Shen Yongliang, Ma Xinyin et al. A Two-stage Identifier for Nested Named Entity Recognition — 2021. — Vol. arXiv:2105.06804. — version 2. Access mode: https://arxiv.org/abs/2105.06804.

[7] Tan Chuanqi, Qiu Wei et al. Boundary Enhanced Neural Span Classification for Nested Named Entity Recognition // Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence. — New York, New York, USA, 2020. — P. 9016–9023.

[8] Tan Zeqi, Shen Yongliang et al. A Sequence-to-Set Network for Nested Named Entity Recognition — 2021. — Vol. arXiv:2105.08901. — version 2. Access mode: https://arxiv.org/abs/2105.08901.

[9] Tong Meihan, Wang Shuai et al. Learning from Miscellaneous Other-Class Words for Few-shot Named Entity Recognition — 2021. — Vol. arXiv:2106.15167. — version 1. Access mode: https://arxiv.org/abs/2106.15167v1.

[10] Waldis Andreas, Mazzola Luca. Nested and Balanced Entity Recognition Using Multi-Task Learning — 2021. — Vol. arXiv:2106.06216. — version 1. Access mode: https://arxiv.org/abs/2106.06216.

[11] Wang Bailin, Lu Wei. Neural Segmental Hypergraphs for Overlapping Mention Recognition — 2018. — Vol. arXiv:1810.01817. — version 1. Access mode: https://arxiv.org/abs/1810.01817v1.

[12] Wang Yu, Li Yun et al. HIT: Nested Named Entity Recognition via Head-Tail Pair and Token Interaction // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. — Punta Cana, Dominican Republic, 2020. — P. 6027–6036.

[13] Yang Yi, Katiyar Arzoo. Simple and Effective Few-Shot Named Entity Recognition with Structured Nearest Neighbor Learning — 2020. — Vol. arXiv:2010.02405. Access mode: https://arxiv.org/abs/2010.02405.

[14] Fei Li, Zhichao Lin, Meishan Zhang, Donghong Ji. A Span-Based Model for Joint Overlapped and Discontinuous Named Entity Recognition. — 2021. — Vol. arXiv: 2106.14373. — version 1. Access mode: https://arxiv.org/pdf/2106.14373.pdf.

[15] Kuratov Yuri, Arkhipov Mikhail. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. — 2019. — Vol. arXiv: 1905.07213. — version 1. Access mode: https://arxiv.org/pdf/1905.07213.pdf.

[16] Guo Zhijiang, Zhang Yan, Lu Wei. Attention Guided Graph Convolutional Networks for Relation Extraction. — 2020. — Vol. arXiv: 1906.07510. — version 8. Access mode: https://arxiv.org/pdf/1906.07510.pdf.

[17] Artemova, Ekaterina and Zmeev, Maksim and Loukachevitch, Natalia and Rozhkov, Igor and Batura, Tatiana and Braslavski, Pavel and Ivanov, Vladimir and Tutubalina, Elena. RuNNE-2022 Shared Task: Recognizing Nested Named Entities // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialog”. — 2022.