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

Moscow, June 15–18, 2022

Contrastive fine-tuning to improve generalization in deep NER

Ivan Bondarenko
Novosibirsk State University
Russia, Novosibirsk
i.bondarenko@g.nsu.ru

Abstract

A novel algorithm of two-stage fine-tuning of a BERT-based language model for more effective named entity recognition is proposed. The first stage is based on training BERT as a Siamese network using a special contrastive loss function, and the second stage consists of fine-tuning the NER as a "traditional" sequence tagger. Inclusion of the contrastive first stage makes it possible to construct a high-level feature space at the output of BERT with more compact representations of different named entity classes. Experiments have shown that this fine-tuning scheme improves the generalization ability of named entity recognition models fine-tuned from various pre-trained BERT models. The source code is available under an Apache 2.0 license and hosted on GitHub https://github.com/bond005/runne_contrastive_ner

Keywords: named entity recognition, contrastive learning, Siamese neural networks, BERT
DOI: 10.28995/2075-7182-2022-21-70-80

Сопоставительное дообучение для повышения обобщающей способности нейросетевого распознавателя именованных сущностей

Иван Бондаренко
Новосибирский государственный университет
Россия, Новосибирск
i.bondarenko@g.nsu.ru

Аннотация

Предложен новый алгоритм двухэтапного дообучения нейросетевой языковой модели BERT для более эффективного распознавания именованных сущностей. Первый этап представляет собой дообучение BERT как Сиамской нейронной сети с использованием специальной сопоставительной функции потерь, а второй этап связан с окончательным дообучением распознавателя именованных сущностей как "традиционного"классификатора элементов последовательности. Добавление первого этапа, основанного на методе сопоставительного обучения, обеспечивает построение высокоуровневого признакового пространства на выходе нейросетевой языковой модели BERT с более компактными представлениями разных классов именованных сущностей. Эксперименты показывают, что такая схема дообучения повышает обобщающую способность распознавателей именованных сущностей на базе целого ряда предобученных языковых моделей BERT. Исходный код доступен под лицензией Apache 2.0 и размещен на GitHub https://github.com/bond005/runne_contrastive_ner

Ключевые слова: распознавание именованных сущностей, сопоставительное обучение, Сиамские нейронные сети, BERT

1 Introduction to problem: why and what do we solve?

The NER (Named Entity Recognition) task has been known for a long time and is generally formulated as finding key elements, like names of people, places, brands, monetary values, and more, in a text. This is used in many software products, so there is a lot of research on this topic. Specifically, over the past few years, most NER solutions have been based on the Transformer architecture.

There are many different approaches to Transformer fine-tuning. First, there is a development direction dedicated to the modification of the loss function and a specific problem statement. For example, training problem could be set as machine reading comprehence (question answering) instead of the standard sequence classification, or focal loss, dice loss and other things from other deep learning domains could be used instead of the standard cross-entropy loss function. Second, there are papers devoted to BERT extension, related to adding more input information from the knowledge graph, morpho-syntactic parsers and other things. Third, there is a group of algorithms associated with changing the learning procedure, such as metric learning (contrastive learning).

Each direction has its own advantages and disadvantages, but the metric learning seems the most promising to us. Because the goal of any training is not to overfit the training sample and not just to take the top of the leaderboard on a particular test sample from the general population, but to ensure the highest generalization ability on the general population as a whole. High generalization ability is associated with good separation in the feature space. A good separation is possible when objects of different classes form sufficiently compact regions in our space. And methods of contrastive learning achieve better separation.

Our goal is to test, on the basis of the RuNNE competition (Artemova et al., 2022), how true are these theoretical considerations in practice and how much will the use of comparative learning in BERT's fine tuning allow us to build more compact high-level representations of different classes of named entities and, as a result, improve the quality of recognition of named entities.

2 Standing on the shoulders of giants: research of our predecessors

Let's consider some important approaches illustrating the previously described directions. In one the approaches, described in the paper "A Unified MRC Framework for Named Entity Recognition"(Li et al., 2020a), the sequence tagging task transformed into question answering. The inputs of the model are the "question" in natural language (for example, "Find locations in the text") and the text, for which the model must predict the indexes of the beginning and the end of the entity, which is the answer to the "question". Questions are generated separately for each entity class and can be done manually or based on rules. The model trained in this way showed the best quality of results compared to other BERT-based models on various datasets, including those with nested entities.

A comparable result was achieved by the WCL-BBCD(Zhou et al., 2022) approach, where, instead of changing the task, a modified training procedure of the BERT-based model is used. Additionally, a priori information from knowledge graphs is also used. The idea of BERT training is representation learning - it is necessary to teach the model to separate representations of different classes in the feature space. To do this, the generation of "similar" sentences by translating into another language and then back-translating into the original language is used. The better the model determines whether a pair of sentences are similar or not, the more the classes of entities contained in these sentences are separated in the feature space, and the better the model will be fine-tuned for the NER task. This idea is similar to ours.

Contrastive learning also helps to get better results in the few-shot NER problem. For example, the authors of the CONTaiNER(Das et al., 2021) article successfully used contrastive learning to solve the few-shot problem and surpassed the results of previous models. In that model, unlike ours, Gaussian embeddings were used and there was no second stage of fine-tuning.

We looked at the approaches that, in one way or another, inspired us to create our model. There are also many other popular and interesting approaches to various formulations of the NER problem.

3 CoNER: proposed contrastive named entity recognizer

We propose **CoNER** - a **Co**ntrastive **N**amed **E**ntity **R**ecognizer. It is based on a special two-stage fine-tuning of a pretrained BERT language model:

- The first stage is a fine-tuning of the pretrained BERT as a Siamese neural network to contrast semantics of different entities in text pairs
- The second stage is a fine-tuning of the resultant neural network as a standard NER (i.e. sequence classifier) with a BILOU tagging scheme

3.1 Contrastive fine-tuning

The first stage of the fine-tuning is based on working with BERT as with the Siamese neural network with a contrastive loss (see Figure 1).

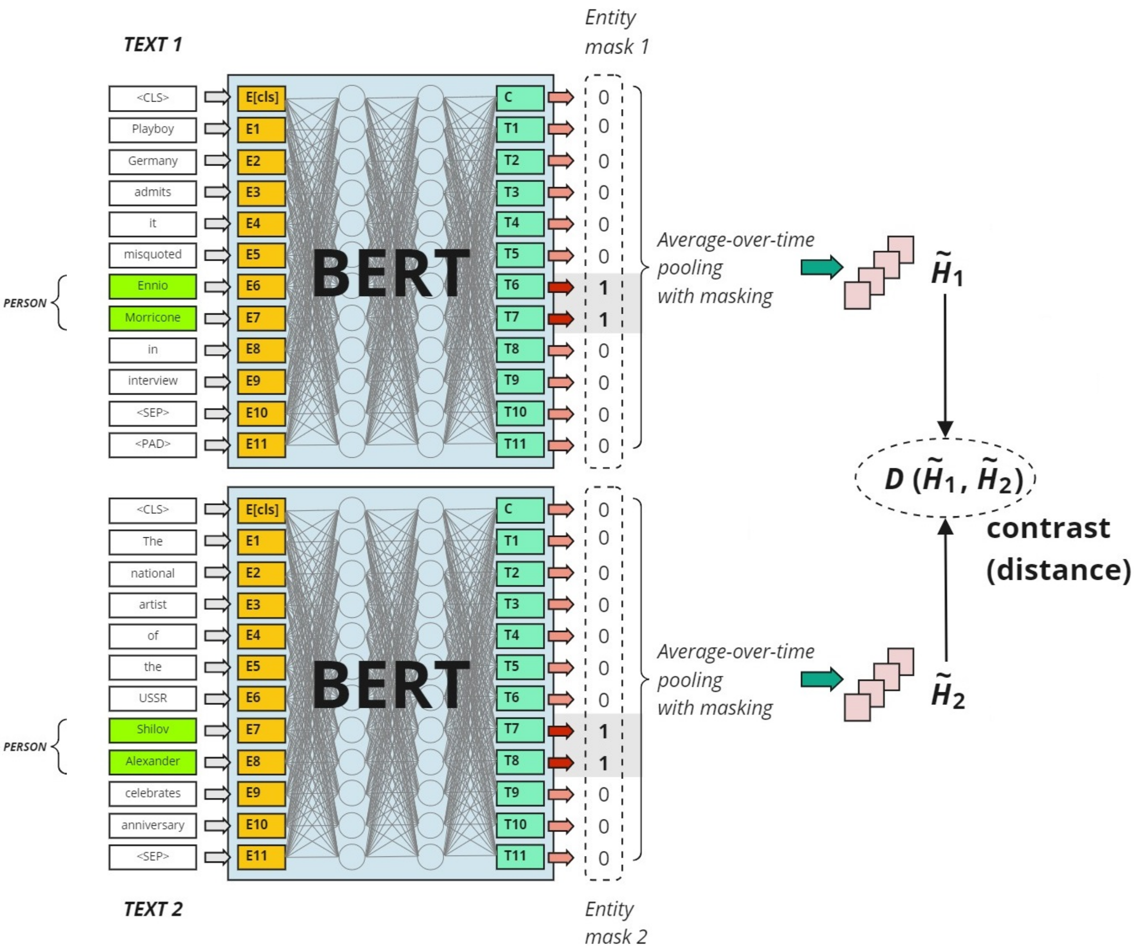


Figure 1: Entity masking example for the sequence outputs of two BERT sub-networks in the contrastive learning with Siamese BERT.

Unlike the well-known Sentence BERT (Reimers and Gurevych, 2019), we work with pairs of entities such as text segments instead of pairs of whole texts. Each text segment is specified as a pair $Ent =< S, M >$, where S corresponds to a tokenized sentence containing the entity, and M specifies a token-in-text mask that defines the bounds of the entity. Thus, the named entity embedding generated from the last hidden layer of BERT is described using the following expression:

$$H(S, M) = F_{\mathrm{BERT}}(S) \circ M, \tag{1}$$

where ($\circ$) is a masking operator which implements an element-wise multiplication between matrices of the same sizes (a masked matrix of embeddings and a masking matrix of zeros and ones).

After that we apply L_2 normalization to the named entity embedding to regard it as point on a unit hypersphere:

$$\widetilde{H}(S, M) = \frac{H(S, M)}{||H(S, M)||} \tag{2}$$

Finally, we formulate the probability of the classes of the entities in the pair matching based on the euclidean distance between their embeddings as points on the unit hypersphere:

$$p(\widetilde{H}_1, \widetilde{H}_2) = \frac{1 + \exp(-m)}{1 + \exp\left(D(\widetilde{H}_1, \widetilde{H}_2) - m\right)}, \tag{3}$$

where m is the margin parameter to inflict of a penalty on the matched pairs with a too large distance (usually it equals to 1). Then we can use the log-loss as in the classification case. This distance based logistic (DBL) loss for Siamese neural networks was firstly proposed for a special computer vision task, concerned with localizing street views on satellite images. (Vo and Hays, 2016)

In comparison with a "classical" contrastive loss function, which is popular for Siamese neural networks, the DBL loss function is more effective owing to quicker convergence. In contrast to classification loss function after Siamese network structure such as Sentence-BERT, the DBL loss function does not need an additional trainable layer what allows us to stay focused on fitting the BERT sub-network only. Unlike the popular N-pairs loss and the SupCon loss function (Khosla et al., 2020), the DBL loss function provides a more stable training process on small mini-batches, and it is less inclined to the exploding gradients problem in such situations.

3.2 Final fine-tuning for NER

We use the BERT model fine-tuned according to the principles from subsection 3.1 as the base for a multi-head sequence classifier where each head corresponds to one of 29 named entity classes. Since a named entity can consist of several tokens, then we use the **BILOU** tagging scheme and describe the named entity class as a system of five token classes: the **O**utside, the **B**eginning, the **I**nside and the **L**ast tokens of multi-token chunks (see example 5) as well as **U**nit-length chunks (see example 4).

For example, the text "*В Китае отметили 170-летие публикации «Коммунистического манифеста»*" (in English, "China celebrated the 170th anniversary of the publication of the Communist Manifesto") contains several named entities of different classes. Examples 4 and 5 illustrate applying BILOU tagging to the named entities in this text, which consist of single token and multiple tokens, accordingly:

(4)	*В*	*Китае*	*отметили*	*170-летие*	*публикации*	*«*	*Коммунистического*	*манифеста*	*»*
	O	**U**	O	O	O	O	O	O	O

‘BILOU labeling for the unit-length entity of the COUNTRY class’

(5)	*В*	*Китае*	*отметили*	*170-летие*	*публикации*	*«*	*Коммунистического*	*манифеста*	*»*
	O	O	O	O	O	**B**	**I**	**I**	**L**

‘BILOU labeling for the multi-token entity of the WORK_OF_ART class’

We have two reasons for using BILOU instead of well-known BIO (the Beginning - the Inside - the Outside) tagging scheme:

1. some preceding experiments demonstrate that the BILOU formalism outperforms the BIO tagging scheme. (Ratinov and Roth, 2009)
2. but also more importantly, the BILOU scheme brings more a priori knowledge about the structure of a natural language in trainable model.

As a result of the above, the second fine-tuning stage is defined as training to solve 29 tasks of 5-class token classifications. The trainable algorithm consists of:

- the shared hidden layer (Transformer base) fine-tuned on the previous stage
- 29 different time-distributed dense layers (neural heads) initialized randomly.

The total loss function for the trainable algorithm is formulated as the sum of particular loss functions (named entity losses) associated with each of 29 neural heads. In this case the key problem is determining the named entity loss function. We propose a special loss as weighted combination of the usual categorical crossentropy for multiclass classification and the dice loss for binary classification ***O* vs. *non-O*** (i.e. all entity tokens including the Beginning, the Inside, the Last, and the Unit are opposed to the Outside):

$$NEL = -\alpha \cdot \sum_{i \in T} (\mathbf{y}_i \cdot \log(\mathbf{p}_i)) - \frac{2 \cdot \sum_{i \neq O} \mathbf{p}_i \cdot \sum_{i \neq O} \mathbf{y}_i + \gamma}{\left(\sum_{i \neq O} \mathbf{p}_i\right)^2 + \left(\sum_{i \neq O} \mathbf{y}_i\right)^2 + \gamma} + 1.0 \tag{6}$$

where $T = \{O, B, I, L, U\}$ is the set of all BILOU tags for the named entity, α is the weight of the cross-entropy item, and γ is the smoothing factor in the nominator and denominator of the dice loss.

The dice loss item is included in the total formula of the named entity loss to reduce the influence of the background-object label imbalance in data (evidently, frequency of the Outside tag is far greater than frequency of all entity tags, and it is a severe issue). As is well known, the dice loss has the class re-balancing property (Li et al., 2020b). Thus, the proposed named entity loss combines two advantages:

- the dice loss item attaches the robustness to imbalance,
- the categorical cross-entropy loss item accounts for the BILOU scheme which leads to better modeling the multi-token entities.

It should be clarified that the first advantage is important for the named entity recognition task, since labels of any named entity class are very unbalanced (the number of words labeled as entities is extremely less than number of non-entity words). However, the dice loss conforms to the binary classification problem, while the BILOU scheme can be implemented as the multiclass classification problem only. Consequently, we need to preserve the multiclass component in the formulated loss function. This implies the significance of the second of these advantages.

3.3 Rescoring with Viterbi algorithm

The posterior probability distribution $P(L|W)$ of the BILOU tag sequence L given the input word sequence W is calculated using Bayes' theorem:

$$P(L|W) = \frac{P(W|L) \cdot P(L)}{P(W)}, \tag{7}$$

where $P(W|L)$ is an observation likelihood estimated by the corresponding neural head of the multi-head NER from subsection 3.2, $P(L)$ is a prior distribution of BILOU tags determined by the language structure, and $P(W)$ is a marginal distribution which does not depend on any BILOU tag sequence (hence we can neglect it). Thus the best way to find an optimal sequence of the BILOU tags is based on well-known Viterbi search:

$$L^* = \arg\max_L \left(p(W|L)P(L)\right). \tag{8}$$

The description of Viterbi algorithm is trivial and can be found in various papers and books on natural language processing and speech recognition (for example, see (Jurafsky and Martin, 2008)). Nevertheless, it would be interesting to describe a technique for forming the prior distribution of BILOU tags. The fact is that the input sequence does not consist of true words. Text units of the sequence are sub-words built with a byte pair encoding according to the tokenization algorithm for the BERT model. Each true word can consist of one or multiple such sub-words. Correspondingly, we define four cases:

- the sub-word is equal to the true word,
- the sub-word is the first part in the true word;
- the sub-word is the middle part in the true word;
- the sub-word is the last part in the true word.

These cases are illustrated by the example 9. The true word "*импичмент*" (in English, "impeachment") is represented by one sub-word, but another true word "*Руссефф*" (in English, "Rousseff", which is the surname of the 36th president of Brazil) consists of three sub-words, i.e of the first, middle and last parts of the true word, accordingly.

(9)	*Президенту*	*Бразилии*	*Дильме*	*Руссефф*	*грозит*	*импичмент*
	президенту	брази ##лии	ди ##л ##ме	рус ##сеф ##ф	грозит	импичмент

‘Possible tokenization cases (the SberDevices RuBERT-large tokenizer is used)’

Possibility transitions from one BILOU tag to another are different for each of these cases, and the prior distribution $P(L)$ is defined differently too. The discrete-time Markov chains for these four cases are shown on Figure 2.

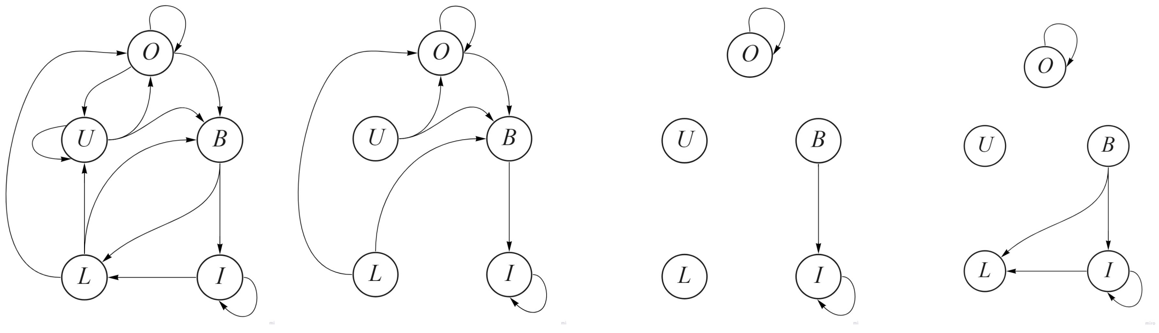


Figure 2: Four discrete-time Markov chains, which describe the BILOU tag transitions for: (a) the sub-word equals the whole word, (b) the first, (c) the middle, and (d) the last sub-word in the word.

Using the Viterbi search instead of the simplest greedy search allows us to rescore the output probabilities of the neural head accounting for some a priori knowledge, and the abovementioned way to determine the prior distribution of BILOU tags rationally specifies this knowledge.

4 Analysis and discussion

We evaluated the quality of several versions of CoNER on the NEREL dataset (Loukachevitch et al., 2021) in the context of the abovementioned RuNNE competition (Artemova et al., 2022). The F1 score is the relevance criterion of the quality, and it allows us to compare different NER algorithms to confirm or deny our hypothesis about effectiveness of the contrastive fine-tuning. Nevertheless, to explain the reasons of effectiveness of the the contrastive fine-tuning in CoNER, we need more than just comparing its F1 score to other NERs. In addition, it is important and necessary to analyze CoNER's mistakes in an attempt to find some patterns and regions of the algorithm errors. These issues will be the subject of further discussion in this section.

4.1 CoNER vs. standard NER: what's better?

We proposed the described two-stage fine-tuning to improve a NER quality. We explained the theoretical basis of the improvment. However, practice is the criterion of truth. We organized a series of experiments to compare two fine-tuning schemes:

- **standard NER**: fine-tuning BERT as a sequence classifier only;
- **CoNER**: two-stage fine-tuning BERT including the contrastive-based learning as the first stage.

Both fine-tuning schemes were started from three pretrained BERT models: the DeepPavlov RuBERT and two variants (base and large) of the SberDevices RuBERT. Here is a brief note about the difference between these pretrained models. The DeepPavlov model was trained on the Russian part of Wikipedia and news data. The SberDevices team added the Taiga corpus and a fiction corpus into the training data for both of its models. The SberDevices RuBERT-large is larger than the other two models. Also, in contrast to the DeepPavlov RuBERT, all the models of SberDevices are lowercased.

According to the description in Section 1, there were two formulations of the problem, or two kinds of tasks: the main task and the few-shot task. All entity classes in the few-shot formulation were very rare in the training data, which led to a significantly greater imbalance of data set in relation to these entities.

Both tasks were solved using the same NER algorithm which was trained on the common training set. After that, the quality of this NER for solving any task was evaluated on the same test inputs, but test labels for each task were different depending on the corresponding entity classes. The results of the experiments in the main formulation (main results) are presented in Table 1, and the results of analogous experiments for the few-shot task (few-shot results) are presented in Table 2.

Type of pre-trained model	Standard NER	CoNER
DeepPavlov RuBERT	0.7202	0.7425
SberDevices RuBERT base	0.6931	0.7233
SberDevices RuBERT large	0.7089	0.7113

Table 1: Main results (F1-macro scores) after different fine-tuning schemes from different pre-trained BERT models on the test data (i.e. at the RuNNE test phase).

Type of pre-trained model	Standard NER	CoNER
DeepPavlov RuBERT	0.3231	0.4037
SberDevices RuBERT base	0.3320	0.5099
SberDevices RuBERT large	0.4372	0.5256

Table 2: Few-shot results (F1-macro scores) after different fine-tuning schemes from different pre-trained BERT models on the test data (i.e. at the RuNNE test phase).

Results of CoNER appear to be better for either type of the task (main and few-shot). We used the Wilcoxon signed-rank test for statistically significant acceptance of this statement. To test the null hypothesis that there is no difference between quality measurements of a standard NER and CoNER, we applied the two-sided test. Evidently, size of both measurement sets is 6 as sum of 3 samples for the main task and 3 samples for the few-shot task. As a result, the test statistic was 0.0 with p-value of 0.03125. Hence, our null hypothesis was rejected at a confidence level of 0.05, and the differences in quality were confirmed.

The following conclusions and summary can be made from these experiments:

1. CoNER is better than a standard NER, and inclusion of contrastive learning in the fine-tuning scheme improves the generalization ability of any NER in any formulation of the problem.
2. This advantage of CoNER is more apparent for the few-shot task.
3. Using case-insensitive pretrained model similar to all SberDevices models reduces the NER quality for the main task. However, it likely is more effective in the few-shot formulation of the problem owing to suppression of the over-fitting on very imbalanced text datasets.

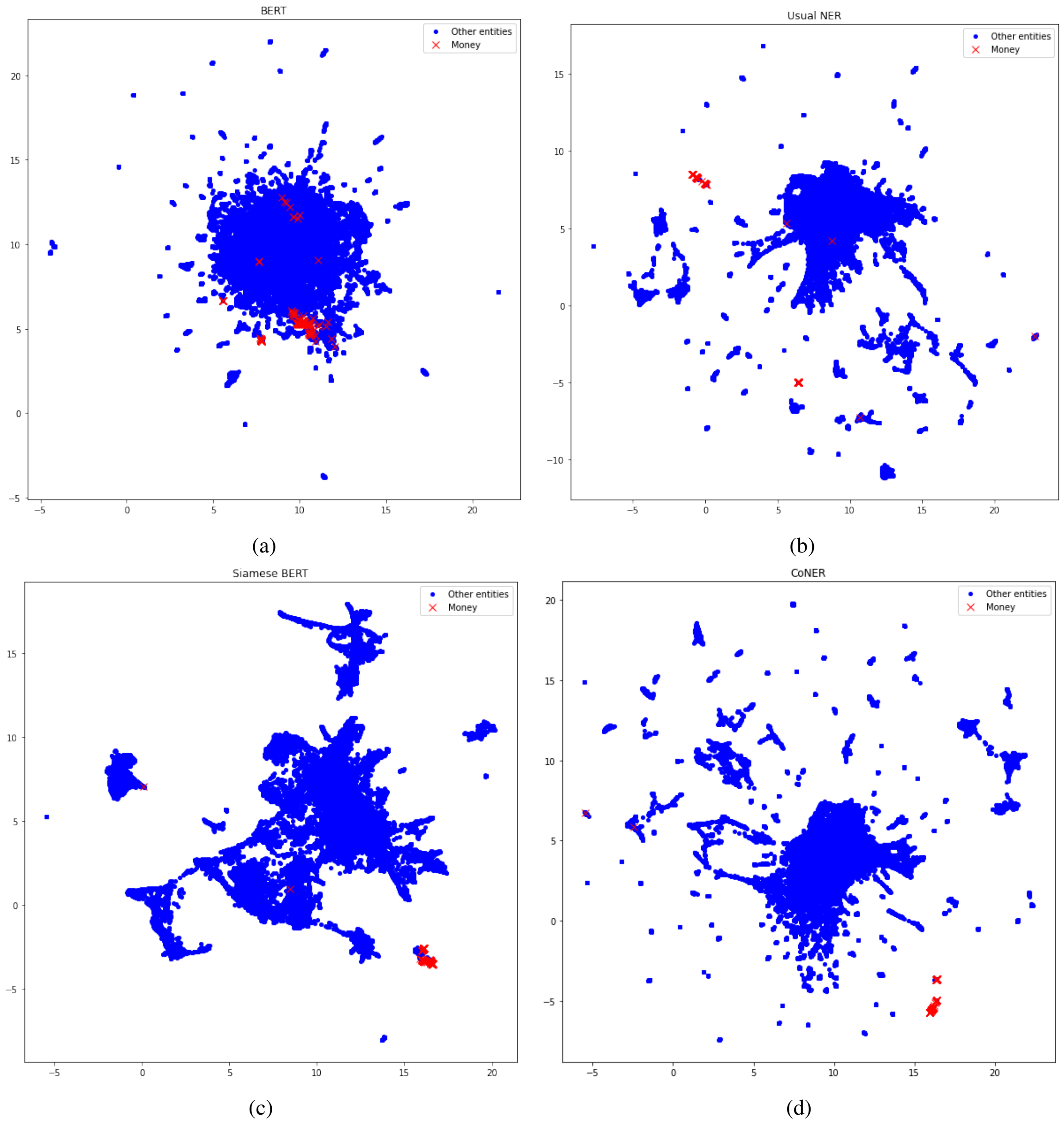
For the final submission to the RuNNE leaderboard, we selected CoNER based on the BERT model pretrained by the DeepPavlov team. This submission turned out to be the third out of 10 in the leaderboard for main task, and the eighth out of 10 in the few-shot formulation of the problem.

4.2 Why does contrastive-based fine-tuning matter?

It can be seen that CoNER is better than a standard NER, and the contrastive fine-tuning works. Nevertheless, we would like to explain these results. In section 2, we proposed a two-stage fine-tuning with the contrastive first stage based on our supposition that Siamese neural network as a typical kind of the contrastive learning algorithm has better discriminative ability in comparison to a standard classifier.

As they say, "a picture is worth a thousand words", and this picture is shown on Figure 3. It illustrates the compactness of the entity word representations in different feature spaces obtained from the sequence output of different BERT models. MONEY is used as a typical example of named entity class in this figure. As Figure 3a demonstrates, entity words and other words cannot be well separated in the usual pretrained BERT embedding space. Figure 3b shows that they have become better separated after fine-tuning BERT as standard sequence classifier. Better, but not by much. And only inclusion of the contrastive learning in the fine-tuning scheme, i.e. fine-tuning BERT as Siamese neural network, significantly increases the compactness and separability of entities in the word embedding space (this effect is clearly visible in Figures 3c and 3d).

Besides the abovementioned visual explanation, we analyzed the quality of representations of entity



(a) (b) (c) (d)

Figure 3: UMAP projections of 768-dimensional contextual embeddings of test words associated with MONEY entities and everything else, where (a) the embeddings before any fine-tuning RuBERT, (b) after fine-tuning as the standard NER, (c) after the first and (d) second stage of the proposed two- stage fine-tuning. The red crosses present words of money entities, and the blue dots are everything else, i.e. words of other entity classes and various background words.

words with and without the contrastive-based fine-tuning stage using the Silhouette Coefficient, well-known as a representative metric for the clustering performance evaluation. We formulated a hypothesis that the contextual word embeddings generated by BERT from CoNER allow us to build a more separable entity space compared to BERT from a standard NER. To test this hypothesis, we did the following steps:

1. We calculated contextual word embeddings from the last hidden output of the standard NER for all words in the test set. Thereby, we built a new feature space of words from named entities.
2. We separated all these words in the formed feature space into two clusters according to manual entity labeling: entity and non-entity. We did it for each of 29 entity classes, and we obtained 29 kinds of clustering.

3. We evaluated the Silhouette Coefficient for each kind of clustering from 29. Thus, we formed 29 measure samples.
4. Then we repeated these three steps with CoNER instead of the standard NER. As a result, we got yet another measure sampling with 29 elements.

The two measurement samplings are shown in Table 3. The superiority of representations from CoNER is observable to the unaided eye. Nevertheless, we applied the dependent t-test for paired samples for statistically significant acceptance of difference between these two samplings (they comes from a normal distribution, and their sizes are too large to effectively apply the Wilcoxon signed-rank test). The test statistic was -2.2866 with p-value of 0.029985. Hence, our null hypothesis that the contrastive-based fine-tuning does not improve separability of entities in the embedding space was rejected at a confidence level of 0.05, and the differences in quality were confirmed. The inclusion of fine-tuning BERT as Siamese neural network matters for BERT-based NER.

Entity class	**Word number in training texts**	**Standard NER**	**CoNER**
AGE	1506	0.4363	0.4507
AWARD	757	0.4959	0.5014
CITY	1432	0.4678	0.4639
COUNTRY	2671	0.4847	0.4788
CRIME	482	0.4064	0.4443
DATE	7087	0.4042	0.3886
DISEASE	**53**	**0.3022**	**0.4220**
DISTRICT	207	0.5490	0.5297
EVENT	5002	0.3102	0.3123
FACILITY	893	0.5049	0.5196
FAMILY	34	0.3242	0.4240
IDEOLOGY	368	0.4638	0.4510
LANGUAGE	45	0.3438	0.4014
LAW	1609	0.4601	0.4750
LOCATION	462	0.5308	0.5331
MONEY	635	0.4982	0.5240
NATIONALITY	434	0.4716	0.4271
NUMBER	1410	0.4745	0.4597
ORDINAL	710	0.5275	0.5456
ORGANIZATION	7127	0.4638	0.4588
PENALTY	**73**	**0.4538**	**0.5251**
PERCENT	216	0.5028	0.5166
PERSON	8063	0.3553	0.3439
PRODUCT	398	0.3594	0.3636
PROFESSION	7970	0.4314	0.4259
RELIGION	101	0.5377	0.5609
STATE OR PROVINCE	485	0.5562	0.5741
TIME	693	0.4993	0.4914
WORK OF ART	**74**	**0.4396**	**0.4796**

Table 3: The Silhouette Coefficient as the quality of different entity classes representation in the feature space generated with CoNER and the standard NER. Entity classes with better value of the Silhouette Coefficient for the standard NER are grey colored. Entity classes for the few-shot formulation of the problem are bolded.

4.3 When does CoNER make errors?

CoNER is good, but not the best according to the RuNNE competition leaderboard. CoNER makes some errors. Confucius said that "*people make errors according to the type of person they are. By observing their errors, you can understand humaneness*". Similarly, by observing algorithm errors, we can understand its generalization ability and some patterns of its work.

Most errors can be divided into two types:

1. **The algorithm does not recognize nested entities of same entity class.** For example, the phrase "*Центральный комитет Коммунистического союза молодёжи Китая*" (in English, "the Central Committee of the Communist Youth League of China") describes the organization, and also it contains three nested organizations and one nested location. Nested organizations are "*Центральный комитет*" (in English, "the Central Committee"), "*Коммунистического союза молодёжи Китая*" (in English, "the Communist Youth League of China"), and "*Коммунистического союза молодёжи*" (in English, "the Communist Youth League"). The nested location is "*Китая*" (in English, "China"). And our CoNER correctly recognizes the "parent" organization and its nested location, but it cannot find any "organization-in-organization".
2. **The algorithm come across an ambiguous manual labeling**. For example, the full text "*пресс-служба филиппинского президента*" (in English, "press service of the Philippine President") was labeled as organization, but the same words "*пресс-служба*" (in English, "press service") were not labeled as a part of the organization entity in the text "*пресс-служба Светлогорского городского суда*" (in Englisn, "press service of the Svetlogorsk City Court").

The first disadvantage is not related to any fine-tuning scheme, and it was determined by the common architecture of proposed solution: entity outputs of neural network were developed without consideration of entity nesting of the same class (we considered that only entities of different classes could be nested). We are going to improve our algorithm on the basis of a special syntactical post-processing of the recognized entity that allows to find nested entities of the same class using noun groups in the "parent" entity.

The second disadvantage is not really significant, because it was explained by the incorrect manual labeling. Furthermore, this effect can demonstrate greater robustness of CoNER, because it does not allow us to discover non-existent patterns by over-adapting to manual labeling.

Also, the algorithm makes a lot of false negative errors for all entities in the few-shot task. This may be explained by the under-fitting effect for very rare entity classes.

5 Conclusion. What's next?

We have confirmed our hypothesis about the contrastive fine-tuning for the NER task. We have also successfully performed error analysis and can apply this to improve our approach.

Further work may include two directions. First, recognition of nested entities of the same class will be implemented (for example, using a special syntactical-based postprocessing). Second, modeling of linguistic concept hierarchy (morphology - syntax - semantics) using a hierarchical multitask learning on both fine-tuning stages will be analyzed, because this technique can increase the generalization ability and reduce the vanishing gradient problem.

Acknowledgements

The work is supported by the Mathematical Center in Akademgorodok under the agreement No. 075-15-2022—282 with the Ministry of Science and Higher Education of the Russian Federation.

Especially the author would like to thank his student and fellow-thinker Alexey Pauls for his help and support.

References

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

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, R. Passonneau, and Rui Zhang. 2021. CONTaiNER: Few-Shot Named Entity Recognition via Contrastive Learning. *Computing Research Repository*, arXiv:2109.07589.

Dan Jurafsky and James H. Martin. 2008. *Speech and language processing - an introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall; 2nd edition, Hoboken, New Jersey, USA.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. // *Advances in Neural Information Processing Systems*, volume 33, P 18661–18673, online. Curran Associates, Inc.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020a. A Unified MRC Framework for Named Entity Recognition. // *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, P 5849–5859, Online. Association for Computational Linguistics.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020b. Dice Loss for Data-imbalanced NLP Tasks. // *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, P 465–476.

Natalia V. Loukachevitch, Ekaterina Artemova, Tatiana Batura, Pavel Braslavski, Ilia Denisov, Vladimir Ivanov, Suresh Manandhar, Alexander Pugachev, and Elena Tutubalina. 2021. NEREL: A Russian Dataset with Nested Named Entities, Relations and Events. // *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP 2021)*, P 876–886, Held Online. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. // *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, P 147–155, Boulder, Colorado, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. // *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, P 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nam N. Vo and James Hays. 2016. Localizing and Orienting Street Views Using Overhead Imagery. // *Proceedings of the 14th European Conference on Computer Vision*, P 494–509, Amsterdam, The Netherlands.

Renjie Zhou, Qiang Hu, Jian Wan, Jilin Zhang, Qiang Liu, Tianxiang Hu, and Jianjun Li. 2022. WCL-BBCD: A Contrastive Learning and Knowledge Graph Approach to Named Entity Recognition. *Computing Research Repository*, arXiv:2203.06925.