Named Entity Normalization with RuBERT for Ending Classification

Shkunkov A. S.
MIPT
Moscow, Russia
shkunkov.as@phystech.edu

Dmitriev D. V.
MIPT
Moscow, Russia
dmitriev.dv@phystech.edu

Abstract

In this paper we describe the results of participation in the Russian Normalization of Annotated Spans shared task (RuNormAS) within Dialogue Evaluation 2021 (Team: shukunkov.a). The shared task included 2 sub-tasks: normalization of named entities and normalization of text spans with different parts of speech. Our system participated only in the first sub-task. We designed a neural network system that classifies the ending of word by its stem. Our experiments showed that using RuBERT model with the context from the nearest words and additional named entity recognition task provides the best result. However, it was not possible to get a result higher than the solution of the authors of the competition with our approach. This paper describes the methodology of our experiments and the results for different models we used.

Keywords: Normalization, Named Entity Recognition, mBERT, RuBERT, Endings

Нормализация именнованных сущностей с помощью RuBERT для классификации окончаний

Шкунков А. С.
МФТИ
Москва, Россия
shkunkov.as@phystech.edu

Дмитриев Д. В.
МФТИ
Москва, Россия
dmitriev.dv@phystech.edu

Аннотация

Эта работа описывает результаты участия в соревновании по нормализации фрагментов текста, которое проводилось на Dialogue Evaluation 2021. Соревнование включало 2 дорожки: нормализация именованных сущностей и нормализация более широкого класса спанов текста, включая нормализацию разных частей речи. В рамках соревнования проводилось участие только в первой дорожке. Нами была разработана система основанная на нейронной сети, которая училась предсказывать окончание слова по его стемме. В результате экспериментов модель, основанная на модели RuBERT с учетом контекста ближайших слов и дополнительной задачей по предсказанию именованных сущностей показала наилучший результат среди остальных опробованных вариантов. Однако, с такой постановкой задачи не удалось получить результат выше решения авторов соревнования. В работе описываются наш подход к решению задаче и результаты экспериментов для использованных моделей.

Ключевые слова: Нормализация, Распознавание именованных сущностей, mBERT, RuBERT, Окончания

1 Introduction

For the present day, pre-trained models like BERT [1] provide state-of-the-results in many natural language processing (NLP) tasks. In particular, named entity recognition (NER) is a common task that nowadays usually solved with such pre-trained models. However, even the best NER models architectures are working worse with morphologically rich languages like Russian. For example, in Russian the text spans "Борис Стругацкий" and "Бориса Стругацкого" represent the same entity type - PERSON, but their spelling is different. This difference leads to a lower NER performance, so we need to deal with a normalization task. It can be formulated as finding a standardized name form for an entity. "Natasha" library provides the most popular tool for normalization in Russian that uses syntax dependencies to produce correct normalization. However, there is still some space for the overall normalization quality improvement. Within the shared task we aimed to study what one can achieve with the BERT language model if the normalization problem is formulated as an ending classification task. During the shared-task we designed several models based on BERT and RuBERT [5] for ending classification. In this work we describe our approach and results of the experiments. We made our code available at `https://github.com/Ryzhtus/ru-norm-as-classification`.

2 Shared-Task Overview

RuNormAS (Russian Normalization of Annotated Spans) is the shared task for normalizing named entities - bringing named entities to their initial form. The task is to normalize certain words from the group without changing the forms of the remaining words. The authors paid attention to the use of the document context, since the initial form for many words can only be determined using the context. The training data contains 2493 documents, and the test data contains 4370. The shared-task is divided into 2 sub-tasks:

2.1 Sub-tasks

1. Normalization of named entities. The goal is to normalize named entities without changing the rest of the words. The data is collected from the articles of the newspaper "Vzglyad". This sub-task takes into account the capitalization.
2. Normalization of a broader class of text spans, including normalization of different parts of speech. The goal of this task is similar to the previous one, but normalization occurs for certain parts of speech. The data is collected from documents of the Ministry of Economic Development.

The quality metric for the task is the percentage of exact matches between the normalization result and the reference.

2.2 Data Description

Each example in train data is divided into 3 documents:

- Text file containing text of the document.
- Annotation file with the beginning and end indexes of the entity in the text. If the entity has breaks, then the beginning and end indexes for each part are written in one line (the parts may be unordered).
- Normalization file where each line has the corresponding normalized version for the entity in annotation file.

3 Solution For Named Entity Normalization

Existing solutions to the normalization problem for the Russian language are usually based on set of rules. In particular, one of these tools for normalization is represented by the Natasha library, which was used by the authors of the competition to obtain a baseline solution. Another rule-based system is described in [8] However, there have been attempts to solve this problem using neural networks. In particular, in [4], a study was conducted on the use of a Sequence-to-Sequence model using Attention to normalize proper names. In our work, it was decided to use a different approach. To solve the Named Entity Normalization task we defined it as the classification task where model learns to predict word's ending by its stem.

3.1 Data Preprocessing

For the formulated problem statement we designed the following preprocessing scheme to obtain train examples for the model:

1. Get the stem of an annotated word using NLTK Snowball Stemmer algorithm [2].
2. Delete the stem from the annotated word to obtain the ending.
3. Delete the stem from the corresponding normalization to extract the remaining part and use it as an ending for normalization.
4. Mark the stem of the annotated word with the tag "<NO>".
5. Mark the ending of the annotated word with the normalized ending from the third step.

Additionally, we cleaned entities from quotation marks, brackets and replaced all entries of "ё" letters with "е" letters. The process described above is illustrated in (Fig. 1a).

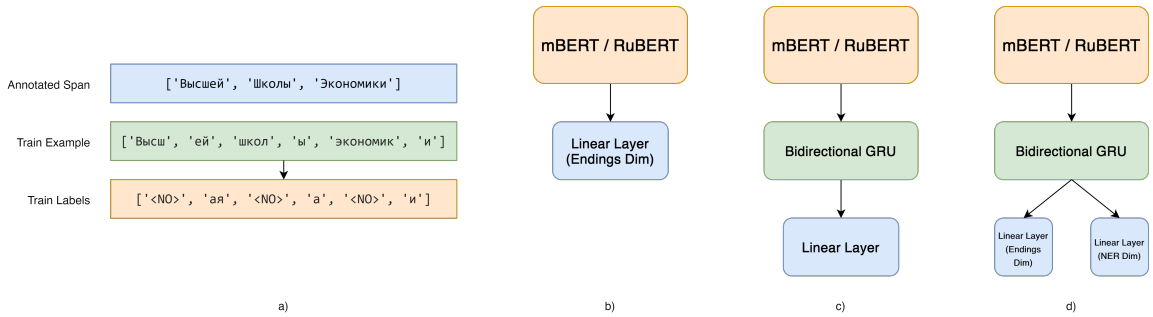

Figure 1: Preprocessing scheme example and Models

For the named entity normalization sub-track within the shared task we built various neural network systems based on one of two models: mBERT [1] and RuBERT [5]. We used base-sized and cased versions from the Hugginface Transformers library [9]. In (Fig. 1b - Fig. 1d) three general model's architectures are presented.

3.2 Basic Model

As a start point we made the simplest model, which has a linear layer for classification on top of the base model. After the training procedure we did a post-processing to get the result for a certain entity. Model's output is a list of endings for each token in input sequence. Because we know the original stem of the particular entity, we concatenate tokens until we won't get the original stem. After this step we took out predictions, filtered them from "<NO>" tags and used the first element of processed tags as the word's ending and concatenated it with the stem. If entity consist of a few words, they were joined by space symbol. The example is provided in (Fig. 2).

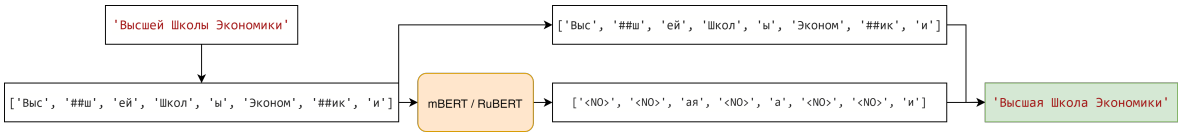

Figure 2: Post Processing example

The following improvement was made by adding Gated Recurrent Unit (GRU) [6] layer between BERT model and the linear layer. These two architectures can be seen in (Fig. 1b) and (Fig. 1c). The results are presented in the Table 1.

3.3 Multi-Task Model

Further, we have tried make our model jointly predict endings with NER tags. This approach is similar to lemmatization with part-of-speech tagging used in works [7], [3]. The idea is that it will allow model to better classify endings for complex entities that consist of 3 or more words and improve prediction

quality for entities that present organizations. Entity tags were obtained during the preprocessing step with Natasha NER tagger. Model's architecture is presented in (Fig. 1d).

3.4 Adding Context to the Basic and Multi-Task Models

The next step that we decided to try was to add some context to our model's input. For that purpose a fixed-size window for additional context was used instead of using a whole sentence. We defined left and right contexts for each entity. Left context consist of a few words from the left, which size is defined by the size of context window. Right context is build in the same way. For our experiments we used 2-size and 3-size context windows. This approach allowed us to significantly improve the model's performance quality on the test subset. In particular, we got the best result with 2-size context window.

4 Results

We summarized all our models results in the Table 1. RuBERT with additional context and Multi-Task RuBERT with additional context models achieved the highest results among the others. One can see that multi-task slightly increases the overall model's prediction quality in the comparison with the model that uses additional context. Nevertheless, multi-task approach gives a robust growth as the improvement for the basic model. We may suggest that though learning NER tags has its profit, but context has a more significant impact on the model for the ending classification task.

Table 1: Final results

mBERT + Linear	0.60951
RuBERT + Linear	0.55846
mBERT + BiGRU + Linear	0.64144
RuBERT + BiGRU + Linear	0.62504
Multi-Task mBERT	0.59319
Multi-Task RuBERT	0.70593
RuBERT + Context (Size 2)	0.76579
Multi-Task RuBERT + Context (Size 2)	0.76797

5 Analysis

We used references for the test subset provided by the authors of the shared-task to analyze the negative predictions from our best model. The test set has 11460 named entities for normalization. Our system provided correct normalization for 8801 example and make 2659 mistakes. We analyzed 2000 mistakes from 2659 and classified them into 4 major categories.

- No parentheses or quotation marks (963 errors). This type of error occurred due to the fact that we removed the quotation marks and brackets during data preprocessing, and therefore the model did not learn them at all. For example, for the reference entity "РИА «Новости»" our system predicts "РИА Новости".
- Wrong ending prediction (847 errors).
- Missing 1 or more than 2 words for a multi-word entity (121 errors). We noticed that this type of error in most cases is typical for entities that designate a place or area. Example (reference: "населенный пункт Счастье", prediction: "Счастье").
- Incorrect capitalization for an entity (64 errors). This type of error is typical for entities consisting of several words with a hyphen between them. Example (reference: "Санкт-Петербург", prediction: "Санкт-петербург"). We analyzed how the Snowball stemmer works with such entities and found that it doesn't break the entity into separate stemms producing a single stemm that includes a hyphen, after which a part of the word goes with a lowercase letter. Example (original entity: "Санкт-Петербурге, result: Санкт-петербург".

Summarizing, we can outline several major problems of our data preprocessing pipeline:

1. The Snowball stemmer doesn't always produce the correct stem for some kind of entities. If the entity is written in 2 or more words with hyphen, then the stem would consist of all words, including hyphen, despite the last one, where the ending will be removed.
2. The other problem is that stemmer removes not only the ending, but also the suffix if the word has it.
3. Filtering brackets and quotation marks from the original entities leads to their absence in predictions for entities that include proper names.

6 Conclusion

In this paper, we presented our model for solving the normalization problem for the Russian language. We formulated the problem as a classification of the ending by the stem of the word. According to the results of the competition, our best model received a result below the baseline presented by the authors of the competition. Based on the results of the work, we can conclude that this problem statement is not suitable for solving such a problem.

References

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). — Minneapolis, Minnesota : Association for Computational Linguistics, 2019. — Jun. — P. 4171–4186. — Access mode: https://www.aclweb.org/anthology/N19-1423.

[2] Bird Steven, Klein Ewan, Loper Edward. Natural Language Processing with Python. — 1st edition. — O'Reilly Media, Inc., 2009. — ISBN: 0596516495.

[3] Kondratyuk Dan. Cross-Lingual Lemmatization and Morphology Tagging with Two-Stage Multilingual BERT Fine-Tuning // Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology. — Florence, Italy : Association for Computational Linguistics, 2019. — Aug. — P. 12–18. — Access mode: https://www.aclweb.org/anthology/W19-4203.

[4] Korvun V. A. Proper Names Normalization Without Semantic Parsing. — 2019. — Access mode: http://www.dialog-21.ru/digest/2019/student/.

[5] Kuratov Yuri, Arkhipov Mikhail. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. — 2019. — 1905.07213.

[6] Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation / Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre et al. // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). — Doha, Qatar : Association for Computational Linguistics, 2014. — Oct. — P. 1724–1734. — Access mode: https://www.aclweb.org/anthology/D14-1179.

[7] LemmaTag: Jointly Tagging and Lemmatizing for Morphologically Rich Languages with BRNNs / Daniel Kondratyuk, Tomáš Gavenčiak, Milan Straka, Jan Hajič // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. — Brussels, Belgium : Association for Computational Linguistics, 2018. — Oct.-Nov. — P. 4921–4928. — Access mode: https://www.aclweb.org/anthology/D18-1532.

[8] Popov A. M. Adaskina Y. V. Andreyeva D. A. Charabet J. Moskvina A. D. Protopopova E. V. Yushina T. A. Named entity normalization for fact extraction task. — 2016. — Access mode: http://www.dialog-21.ru/media/3456/popovametal.pdf.

[9] Transformers: State-of-the-Art Natural Language Processing / Thomas Wolf, Lysandre Debut, Victor Sanh et al. // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. — Online : Association for Computational Linguistics, 2020. — Oct. — P. 38–45. — Access mode: https://www.aclweb.org/anthology/2020.emnlp-demos.6.