Joint task learning for relation extraction and named entity recognition

Authors

Institution

In this work we present our system for RuREBus challenge held together with Dialog 2020 conference. The task consisted of 3 tracks: named entity recognition, relation extraction with provided named entity tags and end-to-end relation extraction. Our system took the first place in the named entity recognition track and the second place in the third track. For the second task we failed to submit the solution till the deadline but it was among the best systems. The systems for all tasks are based on Transformer models.

Key words: relation extraction, named entity recognition, transformer, bert

Совместное обучение моделей для извлечения отношений и именованных сущностей

Авторы

Организация

В данной работе мы представляем нашу систему для соревнования RuREBus, проводящегося совместно с конференцией Dialog 2020. Задача состояла из 3 дорожек: распознавание именованных сущностей, классификация отношений между заранее аннотированными именованными сущностями и извлечение отношений из неаннотированного текста. Наша система заняла первое место в задаче распознавания именованных сущностей и второе место на третьей дорожке. Для второй задачи мы не успели своевременно представить решение, но оно оказалось в числе лучших систем. Системы для всех задач основаны на моделях Transformer.

Ключевые слова: извлечение отношений, распознавание именованных сущностей, transformer, bert

1 Introduction

There are many ways to extract information from text. This task is often solved by extracting named entities and classifying relations between them. One of the most popular datasets for this task is TACRED [6] where semantic relations are understood as relations between two pairs of entities.

Nowadays, state-of-the art results for this dataset are achieved by using Transformer-based models [5]. The most advanced models (according to https://paperswithcode.com/sota/relation extraction-on-tacred) use extra training data.

The authors of Matching the Blanks: Distributional Similarity for Relation Learning [1].

Knowledge Enhanced Contextual Word Representations

Among the systems that make use of only the provided data, the best results were achieved by Joshi et al. [3].

Unfortunately, such annotated datasets are scarce for most languages besides English. Some researchers have tried to solve this problem for the Russian language. They have used unsupervised approaches based on knowledge databases such as Wikidata and online encyclopedias such as Wikipedia. Models trained this way tend to be not specialized because the original database does not contain relations from the required domain. They also tend to work only for the most popular relation types such as geographical or professional ones which are common to Wikipedia.

There are few annotated datasets for the Russian language. Among similar tasks to relation extraction there was held FactRuEval 2016 within the conference Dialog 2016. Within the competition contestants had to extract facts from news articles and to fill special slots in these facts (e.g. one of the fact types was 'Occupation' and its fields were 'POSITION', 'WHO', 'WHERE' and 'PHASE').

RuREBus competition was devoted to the problem of relation extraction and named entities recognition in a specialized business domain.

2 Shared task overview

The organizers of the competition have provided 188 annotated texts as the training dataset and 544 texts as the test dataset for the first and thirds tracks and <N> <N> for the second track respectively. All texts were provided by the Ministry of Economic Development of the Russian Federation. The corpus consists of various regional and strategic plan reports. There are in total 8 named entity classes and 11 semantic relation classes (see Tables 1 and 2). The organizers have also provided a large unannotated dataset for language model fine-tuning. However, we did not use it. A named entity can consist of several words. All entities and relations do not span across sentences.

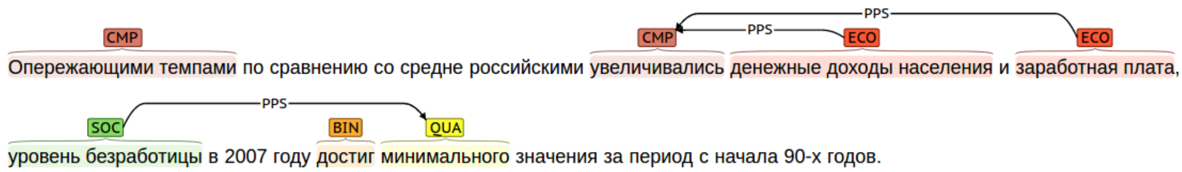

Figure 1: RuREBus annotation example.

Named entity groups could contain rather broad types of entities, for example "SOC" entities contained social groups as well as various social attributes - phrases like 'blue collar workers' and 'housing accessibility' corresponded to this group.

Type	Description
MET	Some quantitative metric
ECO	An economy entity or facility
BIN	A binary attribute
CMP	Comparative attribute
QUA	Qualitative attribute
ACT	Activity, actions, implemented policies
INST	Institutions and organizations
SOC	Social groups and characteristics

Table 1: Named entity types

3 Our solution

The data for the competition was presented in brat format [4] where texts were given as plain txt files and annotations were provided in another file with mixed labels for named entities and relations between them. Thus, we first had to separate the labels and transform the data into special formats used by our models.

Group	Type	Description
Current state of affairs	NNG	now negative
Current state of affairs	NNT	now neutral
Current state of affairs	NPS	now positive
Results	PNG	past negative
Results	PNT	past neutral
Results	PNS	past positive
Forecasts	FNG	future negative
Forecasts	FNT	future neutral
Forecasts	FNS	future positive
Goals	GOL	some abstract goals
Tasks	TSK	tasks and performed actions to achieve goals

Table 2: Semantic relation types

3.1 Named entities recognition

The first task was to annotate named entities. First we transformed the data into the CONLL-2003 format where each line contained a word and its named entity tag. Sentences were separated with newlines. All texts were united in a single file where individual texts were divided with two empty lines. We used the same train, validation and test split as was provided by the organizers. We used a BERT-based system [2] with PyTorch model code and weights provided by Hugging Face. Due to competition being in Russian, we used the multilingual base BERT model.

BERT is a Transformer based model [5]. On top of BERT outputs we added a linear layer with softmax activation function and dropout regularization. The cross entropy loss function was used to train the model.

Our system with 0.561 micro F1-score on the public leaderboard outperformed solutions presented by other contestants.

3.2 End-to-end relation extraction

For end-to-end relation extraction we went with a two-stage approach. At first we used the model from the first track to label named entities. After that using the provided named entities we trained our model to predict semantic relations.

Our relation extraction pipeline was based on the work by Joshi et al. [3]. However, we did not follow the language model fine-tuning procedure described in the paper. We considered the task of relation extraction as a simple classification problem where given a sentence and a pair of named entities we need to correctly choose one of the possible labels for the semantic relation between them. Because we did not use the pre-training procedure by the authors of SpanBERT, our system is equivalent to a plain BERT-model with a softmax layer on top of it. Dropout regularization was used here as well.

The data was transformed into the format used by TACRED dataset [6]. Each relation and its corresponding entities were considered as a single training sample. We also needed to generate pairs that did not contain any relations. Thus, we randomly sampled words that had no relations between them out of sentences.

Akin to BERT-multitask learning, in this competition we wanted to experiment with simultaneous finetuning for separate tracks. RuREBus competition provided an excellent framework for this idea because we had separate tacks with different target values but the same input data.

The system showed 0.132 micro F1-score and it would have taken the first place among the provided systems, if we had managed to submit our solution before the deadline.

3.3 Relation classification with provided named entity tags

This track was very similar to end-to-end relation extraction. However, instead of using named entity labels predicted by our model, we could use the manual annotation provided by the organizers of the competition.

The model for this track is equivalent to the system used for end-to-end relation extraction. We also attempted at using the multi-task learning procedure described in previous section. However, as in the previous case the quality deteriorated when the model was trained to predict named entity tags. Thus, the loss coefficient for named entity recognition was also set to zero.

4 Results

All in all, our named entity recognition model with micro F1-score equal to took the first place in the competition. However, the results are lower than for other named entity recognition datasets (e.g. for the Ontonotes dataset Transformer-based models usually get > 0.85 in F1-score [1]). It can be explained by the small amount of training examples and complexity of the domain.

Our end-to-end relation extraction model was much worse than the model that used manual annotations provided by the organizers.

Multi-task learning did not improve our results for this task.

5 Conclusion

In this work we present our system for RuREBus challenge held together with Dialog 2020 conference. The task consisted of 3 tracks: named entity recognition, relation extraction with provided named entity tags and end-to-end relation extraction. Our system took the first place in the named entity recognition track and the second place in the third track. For the second task we failed to submit the solution till the deadline but it was among the best systems. The systems for all tasks are based on Transformer models.

6 Acknowledgments

We would like to thank the organizers of the competition. We believe that their work will be very helpful for the development of natural language processing for the Russian language.

References

[1] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the Blanks: Distributional Similarity for Relation Learning. In *arxiv.org*, pages 2895–2905, 2019.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. oct 2018.

[1] see http://docs.deeppavlov.ai/en/master/features/models/ner.html

[3] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving Pre-training by Representing and Predicting Spans. 2019.

[4] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations Session at EACL 2012*, Avignon, France, 2012. Association for Computational Linguistics.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 5999–6009, 2017.

[6] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. Position-aware attention and supervised data improve slot filling. In *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 35–45, 2017.