Named Entity Recognition

Jakob Božič
University of Ljubljana
Faculty of Computer and Information Science
Večna pot 113, SI-1000 Ljubljana
jakob.bozic@gmail.com

Abstract

Named entity recognition (NER) is used to extract the named entities from the text into the pre-defined categories. Most commonly these categories are at least names, locations, organizations and other, however in various branches of NER different, branch-specific categories are added. State-of-the-art (SOTA) results are, similarly as in other fields of Machine Learning, given by deep (recurrent) neural networks, which have in last years completed the transition from hand-crafted to deep features. In our work we first give a brief overview of recent development of the field with more focus on Slovene language and then use of the current SOTA (RNN) approaches on ssj500k (Krek et al., 2019) dataset. We will also try to take advantage of transfer learning, which could prove very useful given the rather small amount of resources in Slovene.

1 Related work

In our work we focus on more recent, deep learning based approaches, using Recurrent Neural Networks (RNN). A brief overview of history of NER is given in (Yadav and Bethard, 2019), which also contains results of different NER approaches for four different languages and shows that deep learning approaches outperform traditional methods across the board. Extensive study of NER for slovene language has been presented in (Štajner et al., 2013). Authors have used supervised learning and Conditional Random Fields to propose then state-of-the-art system. They evaluated how introduction of lexicons, part-of-speech tags and conjuctions of neighbouring properties effect the system's capability to correctly classify named entities. In their work they showed that using part-of-speech tags can further improve model's capabilites. When enabling all of their improvements, their model achieves 74% precision and 72% recall, however this drops to 63% precision and 59% recall, when using basic model with no part-of-speech tags and no lexicons. ELMO embeddings (Peters et al., 2018) are deep contextualized word representations which model word usage and how the usage varies in different contexts. They are easily adaptable to existing NLP solutions and usually introduce noticeable performance improvements.

2 Data

We train and evaluate our approach on ssj500k (Krek et al., 2019) dataset. It contains approximately 500.000 words, manually annotated on different levels. Roughly half of the dataset is also annotated with named entities, on which we focus our attention. There are in total 4.398 named entities in total, separated in five classes: loc(ation), org(anization), per(son), misc(ellaneous) and der(ivative)-per(son), however we decided to merge person and derivative-person in per, since derivative-person contained very few samples. Due to only half of the corpus being labeled for named entities we only used those sentences with at least one named entity present, since we could not otherwise tell whether the sentence was labeled for named entities or not. In order to avoid or at least minimise the problems associated with class-imbalance (Buda et al., 2017) we use over-sampling for all categories. In particular, we first create one collection of sentences for each tag, a sentence is put in a collection if it contains at least one named entity for that tag. The last 20% of each collection is put in the test set. For each tag, we then over-sample the remaining part, by repeating each sentence as many times, as the quotient of the length of the collection for that and the length of collection for most common tag. This ensures that all tags are represented approximately equally. We could presumably further improve the data by introducing sentences without any named entities.

3 Methodology

Our model consists of three main parts, (i) an embedder, (ii) an encoder, and (iii) a projector. Embedder transforms characters and words to embeddings and then encoder transforms them in representation which projector uses to make final predictions. Different embedders and different encoders are used and compared against each other. We use either pretrained ELMO embeddings (Ulčar and Robnik-Šikonja, 2019) or we train an embedder with embedding dimension of 64. For encoder we use either two layer LSTM or two layer GRU (Cho et al., 2014) network, with hidden dimension of 64, both normal or bidirectional. We tried using greater dimensions, however we observed over-fitting due to the rather small size of train set. Projector is fixed, a simple fully-connected layer which maps outputs from the encoder in final tags.

4 Results

We evaluated previously described variations of main parts of our model. Results are given in Table 4. It is clearly shown that using ELMO as embedder brings overwhelming performance improvements. Except for the first run, it also appears that GRU consistently outperforms LSTM as well that using bidirectional encoder is better. One possible explanation why ELMO brings such significant performance boost is, that it effectively introduces much more training data, since it was pretrained on a very large corpus. Since there are only approximately 4.000 named entities in our train set, the model is susceptible to over-fitting and when using classical embedders we simply did not have enough training data.

Our models performs significantly better as the one from (Štajner et al., 2013), however since the authors did not report how the train and test data was formed and also how they calculated the final average F1-measure, comparison may not be completely accurate. Their best reported model achieved 72% average F1 measure, whereas ours achieves 88% average F1 measure, which represents more than two-fold reduction of error.

5 Discussion

We evaluated various combinations of embedders and encoders for a NER model and showed that using embedder pretrained on larger corpus brings significant performance improvements. Our best model achieves more than two-fold reduction in error compared to the one from (Štajner et al., 2013).

Embedder	Encoder	Bi-directional	Average F1-measure
Classical	LSTM		51.29
Classical	LSTM	✓	48.30
Classical	GRU		49.22
Classical	GRU	✓	49.37
ELMO	LSTM		85.74
ELMO	LSTM	✓	86.43
ELMO	GRU		86.37
ELMO	GRU	✓	**87.74**

Table 1: Evaluation of different combinations of embedders and encoders on ssj500k dataset.

In the future, we will incorporate pretrained encoder in out network and then try to fine-tune is as our training set is rather small. We will also look for new, larger datasets, which we believe could again bring noticeable performance improvements.

References

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. 2017. A systematic study of the class imbalance problem in convolutional neural networks. *CoRR*, abs/1710.05381.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches.

Simon Krek, Kaja Dobrovoljc, Tomaž Erjavec, Sara Može, Nina Ledinek, Nanika Holz, Katja Zupan, Polona Gantar, Taja Kuzman, Jaka Čibej, Špela Arhar Holdt, Teja Kavčič, Iza Škrjanec, Dafne Marko, Lucija Jezeršek, and Anja Zajc. 2019. Training corpus ssj500k 2.2. Slovenian language resource repository CLARIN.SI.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Matej Ulčar and Marko Robnik-Šikonja. 2019. High quality elmo embeddings for seven less-resourced languages.

Vikas Yadav and Steven Bethard. 2019. A survey on recent advances in named entity recognition from deep learning models.

Tadej Štajner, Tomaž Erjavec, and Simon Krek. 2013. Named entity recognition in slovene text. *Slovenščina 2.0: empirical, applied and interdisciplinary research*, 1(2):58–81.