Hungarian Named Entity Recognition using HuBERT and XLM-RoBERTa

Alex Tibor Papp[1], Benedek Sámuel Pósfay[1]
[1]Budapest University of Technology and Economics, Hungary

ABSTRACT Named Entity Recognition (NER), a classical problem within the field of Natural Language Processing (NLP), is a task of information extraction that seeks to locate and classify named entities in text. In recent years, the advancements in the fields of deep learning and transfer learning have increasingly been adopted by NER systems. However, NER applications designed for low resource languages are yet to have been fully explored. Consequently, in this paper we shall attempt to build a NER system appropriate for Hungarian texts. Finally, we evaluate the relative performance of the proposed solutions and further explore future possibilities in the field of NER research.

I. INTRODUCTION

Despite being conceptually simple, NER is not an easy task. The category of a named entity is highly dependent on textual semantics and its surrounding context. Moreover, there are many definitions of named entity and evaluation criteria, introducing evaluation complications. (Mónica Morrero, 2013)

Current state-of-the-art NER systems employ neural architectures that have been pre-trained on language modeling tasks.

Applying these recent techniques to the Hungarian language can be highly valuable, given that annotated resources are scarce, but unlabeled text data is abundant. In this work, we assess several neural architectures using huBERT (Nemeskey, Natural Language Processing Methods for Language Modeling, 2020) (Nemeskey, Introducing huBERT, 2021) and XLM-RoBERTa (Alexis Conneau, 2019) models to the NER task in Hungarian. We aim to facilitate the reproducibility of this work my making our implementations and models publicly available.

II. RELATED WORK

Deep learning methods have been applied to the NER problem with remarkable success, and a few of these systems have proved to be efficient in more than one language at the same time without major modification. Examples of such models are ELMo (Matthew Peters, 2018), OpenAI GPT (Alec Radford, 2018), BERT (Jacob Devlin, 2018), XL-Net (Zhilin Yang, 2019), RoBERTa (Yinhan Liu, 2019) and Albert (Zhenzhong Lan, 2019).

As always in NLP, all the pioneering research above was centered on English. Support for other languages came in two forms: native contextual embeddings or multilingual variations of the models above. Example for the latter is XLM-RoBERTa which was trained on corpora with around 100 languages.

However, there are some results on NER for the Hungarian language as well. A model based on expert rules defined by linguist experts was developed and tested at the Hungarian Research Institute on Linguistics (Gábor, 2002) and a more recent one which we relied on heavily while doing our research called huBERT, which is the first publicly available Hungarian BERT model trained on a nine-billion-token corpus.

III. THE PROPOSED NETWORK ARCHITECTURE

To achieve our goal, we used transfer learning on the pretrained models of huBERT and XLM-RoBERTa. Since both models are based on BERT the architecture of both is a multi-layer bidirectional Transformer encoder. After loading the pretrained models we used the Hungarian NER data available to us to fine tune the model.

IV. IMPLEMENTATION

In this section, we present the training objective, languages, and data we use.

A. DATA

We gathered our data from a publicly available source provided by the Institute for Computer Science and Control. This data is basically large amounts of annotated text which is a silver standard corpus for Hungarian Named Entity Recognition. The corpus has been automatically generated from the Hungarian Wikipedia. Both Hungarian and English datasets were available, we used the Hungarian one.

We loaded the data into a dataframe and dropped some columns so only the tokens and NER tags remained.

After this we plotted the data and realized that by far the most common NER tag is the O one which means it is not any entity meaningful to us.

Since the tokenization procedure and the model both use sequences as inputs instead of single tokens we had to convert the dataset from single tokens to sentences. Basically, we had to convert a 2D array to a 3D array. We had to be very careful that during the conversion the single tokens will have the matching NER tags assigned to them.

After realizing that we most likely will use huggingfaces, we converted the dataframe into a Datasets object, which took an unexpectedly long amount of time.

Finally, we removed all of the empty words from the tokens and NER tags, because they intervene in the tokenization process.

It is important to note that the data preparation phase is the same for both models we used, so there is only one data preprocessing file in our repository which is to be used before using any of the models.

B. TRAINING

After splitting the dataset to a train and test dataset we could begin the training phase.

First we created dictionaries for label to id conversion, since both models require these. After this we loaded the tokenizer from a pretrained HuBERT model. The tokens were maximum 256 characters long and would be filled with padding if they were shorter than this.

The batch size and the amount of epochs were chosen based on the results of similar models. As for the optimizer we created our own optimizer using the transformers package.

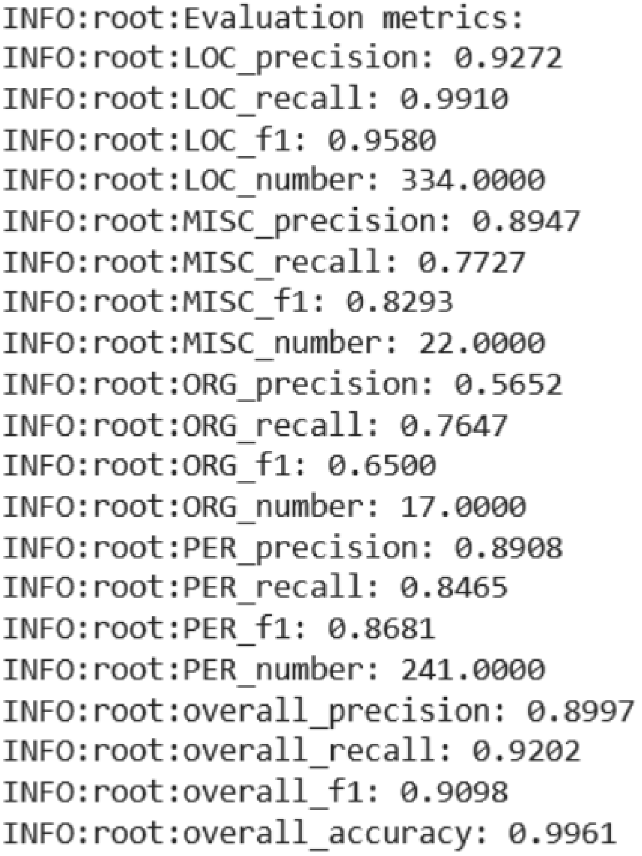
The training of the HuBERT based solution took 16 hours with the usage of GPU's that Google Colab provides. This was when we decided that we need to create another model that's more lightweight.

This was how we stumbled upon the XLM-RoBERTa architecture one of the properties of which is that it is lightweight. Sure enough, the training of 10 epochs took only 50 minutes, orders of magnitude faster compared to our HuBERT based model. Since both our models use the huggingfaces interface, the interchanging of the two models was not a really hard task, all we had to do was change the parameter of the pretrained model calls from HuBERT to XLM-RoBERTa.

C. EVALUATION

For the evaluation we created our own function that returns the precision, recall, F-score and accuracy for every NER tag.

Firstly, I will present the metrics of the HuBERT based model. Of the two models this was of course the better, since it had more than 19 times the amount of learning time. As you can see on the picture, it does very well on Location, Person and Miscellenious entities, but fails to deliver on the Organization entity. This might be because of the low number of Organization entity data. The overall statistics are very convincing, with the overall accuracy being 0.9961. It is of course important to note that most of the tokens are tagged with the O entity. Thus, if the model only predicts the O entity, the overall accuracy will still be very high. That's why it was important for us to create a function that shows the evaluation metrics of all entities.



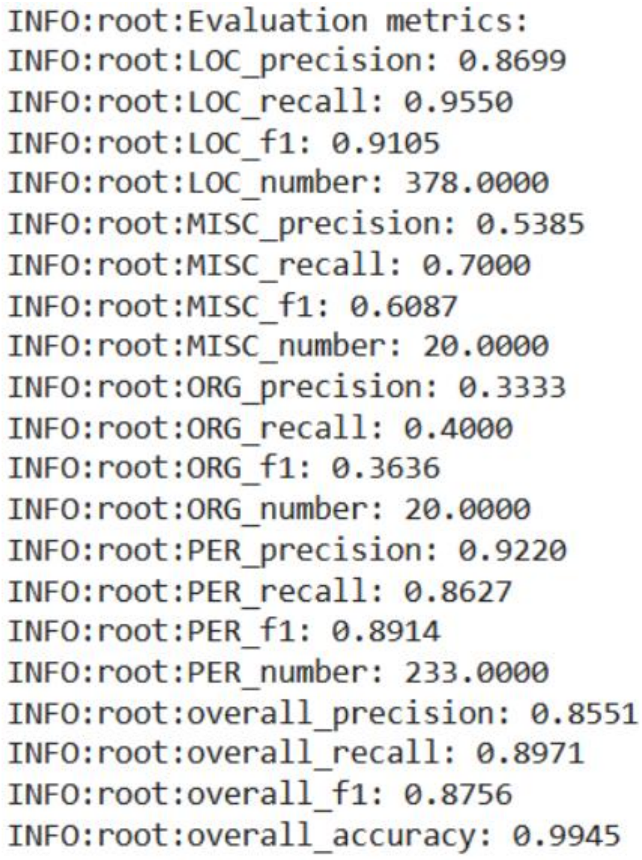
```
INFO:root:Evaluation metrics:
INFO:root:LOC_precision: 0.9272
INFO:root:LOC_recall: 0.9910
INFO:root:LOC_f1: 0.9580
INFO:root:LOC_number: 334.0000
INFO:root:MISC_precision: 0.8947
INFO:root:MISC_recall: 0.7727
INFO:root:MISC_f1: 0.8293
INFO:root:MISC_number: 22.0000
INFO:root:ORG_precision: 0.5652
INFO:root:ORG_recall: 0.7647
INFO:root:ORG_f1: 0.6500
INFO:root:ORG_number: 17.0000
INFO:root:PER_precision: 0.8908
INFO:root:PER_recall: 0.8465
INFO:root:PER_f1: 0.8681
INFO:root:PER_number: 241.0000
INFO:root:overall_precision: 0.8997
INFO:root:overall_recall: 0.9202
INFO:root:overall_f1: 0.9098
INFO:root:overall_accuracy: 0.9961
```

1. Figure: HuBERT based model evaluation

In the evaluation metrics of the XLM-RoBERTa model the difference what learning time makes is visualized. As you can see on the picture below, the model has a pretty good accuracy on the Location and Person entities, but not on the other two. That is because the Organization and Miscellaneous entities have very little data to their name. The overall accuaracy is again very high, albeit lower than the HuBERT based model one, it is of cource a bit misleading due to the reasons discussed above.

D. TESTING

After training the model testing can be easily done by providing an input stream to a huggingface pipeline. The pipeline provides an output with the words and the recognized entities matching them. It also provides a score which shows how confident the model is in the prediction.



```
INFO:root:Evaluation metrics:
INFO:root:LOC_precision: 0.8699
INFO:root:LOC_recall: 0.9550
INFO:root:LOC_f1: 0.9105
INFO:root:LOC_number: 378.0000
INFO:root:MISC_precision: 0.5385
INFO:root:MISC_recall: 0.7000
INFO:root:MISC_f1: 0.6087
INFO:root:MISC_number: 20.0000
INFO:root:ORG_precision: 0.3333
INFO:root:ORG_recall: 0.4000
INFO:root:ORG_f1: 0.3636
INFO:root:ORG_number: 20.0000
INFO:root:PER_precision: 0.9220
INFO:root:PER_recall: 0.8627
INFO:root:PER_f1: 0.8914
INFO:root:PER_number: 233.0000
INFO:root:overall_precision: 0.8551
INFO:root:overall_recall: 0.8971
INFO:root:overall_f1: 0.8756
INFO:root:overall_accuracy: 0.9945
```

2. Figure: XLM-RoBERTa based model evaluation

V. SUMMARY AND CONCLUSION

We are very proud of what we have done considering this is the first time we have used artificial intelligence. To create a program that recognizes entities, especially Locations and Person so well was unimaginable for us half a year ago. What surprised us is that we struggled as much if not more with the preparations of the data as with the training and evaluation.

As for our future plans, there is definitely room for improvement. We are aware that our hyperparameter optimization was not the best and that the transfer learning can still be perfected. We are also thinking of adding document context into our model, which has shown great promise in other models.

Besides this we are both very interested in trying out different NLP applications, especially in the Hungarian language since according to our experience and research there is still a lot of untapped potential in this area.

I. REFERENCES

Alec Radford, K. N. (2018). Improving language understanding with unsupervised learning. *Technical report, Technical report, OpenA*.

Alexis Conneau, K. K. (2019). Unsupervised Cross-lingual Representation Learning at Scale. *arXiv:1911.02116*.

Gábor, K. H. (2002). Nylt tokenosztályok reprezentációjának technológiája. *Technical report, Academys Research Institute for Linguistics, Hungary,*, IKTA-00037/2002.

Jacob Devlin, M.-W. C. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. . *Computing Research Repository, arXiv:1810.04805*.

Matthew Peters, M. N. (2018). Deep contextualized word representations. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers),* 2227-2237.

Mónica Morrero, J. U.-C.-B. (2013). Named Entity Recognition: Fallacies, Challenges and Opportunities. *Computer Standards & Interfaces*, 35(5):482–489.

Nemeskey, D. M. (2020). *Natural Language Processing Methods for Language Modeling*. Eötvös Loránd University.

Nemeskey, D. M. (2021). Introducing huBERT. In *In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2021)* (old.: 3-14). Szeged.

Yinhan Liu, M. O. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692.*

Zhenzhong Lan, M. C. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942.*

Zhilin Yang, Z. D. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. . *arXiv preprint arXiv:1906.08237*.