Priberam Labs at the 3rd Shared Task on SlavNER

Pedro Ferreira, Rúben Cardoso, Afonso Mendes
Priberam Labs / Portugal
{pedro.ferreira, ruben.cardoso, amm}@priberam.pt

Abstract

This document describes our participation at the 3rd Shared Task on SlavNER, part of the 8th Balto-Slavic Natural Language Processing Workshop, where we focused exclusively in the Named Entity Recognition (NER) task. We addressed this task by combining multilingual contextual embedding models, such as XLM-R (Conneau et al., 2020), with character-level embeddings and a biaffine classifier (Yu et al., 2020). This allowed us to train downstream models for NER using all the available training data. We are able to show that this approach results in good performance when replicating the scenario of the 2nd Shared Task.

1 Introduction

This document describes our participation at the 3rd Shared Task on SlavNER, part of the 8th Balto-Slavic Natural Language Processing Workshop, held in conjunction with the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL). It includes three different subtasks: Named Entity Recognition (NER), including detection and classification, lemmatization, and cross-lingual entity linking. The differentiating feature of this shared task is the focus on six Slavic languages: Bulgarian (BG), Czech (CS), Polish (PL), Russian (RU), Slovene (SL), and Ukrainian (UK).

We focus our participation exclusively on the task of NER, and we base ourselves on recent developments that show that cross-lingual embeddings produce good results for a wide range of languages and tasks (Pires et al., 2019; Hu et al., 2020).

Our overall approach is heavily based in Yu et al. (2020), and uses both contextual and character-level embeddings as input to a sequence of models which culminates in a biaffine classifier. Due to the multilingual nature of this task we explore multilingual contextual embedding models, such as Multilingual BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020), as a way of leveraging all the available training data at once. It differs from the approaches presented in the previous edition of the shared task (Piskorski et al., 2019) in the following aspects: (i) we explore more contextual embedding approaches; (ii) we further finetune the contextual embedding model while training the downstream model; (iii) we use the same topics in our train and development sets; and (iv) we use a different classifier architecture.

Our approach resulted in strong performance when replicating the scenario of the 2nd Shared Task on SlavNER.

2 Related Work

Named Entity Recognition corresponds to the task of both finding and classifying named entities in text. Some of the most common approaches to tackle NER that make use of neural networks involve combining models such as Conditional Random Fields (CRFs, Lafferty et al. 2001), bidirectional Long-Short-Term-Memory Neural Networks (biLSTMs, Schuster and Paliwal 1997), and Convolutional Neural Networks (CNNs, LeCun et al. 1989). Two examples of such approaches are Chiu and Nichols (2016) and Ma and Hovy (2016), which use biLSTMs and CNNs to build both word- and character-level features, diverging in the fact that the latter performs decoding with a CRF, while the former uses a linear layer.

With the development of better pre-trained embedding models, transfer-learning became a significant part of approaches that tackle NER. This technique implies using a pre-trained model in order to obtain an embedding for each word of the input. These representations are then used as input to a model that is able to solve the desired downstream task. In terms of pre-trained embeddings we

Proceedings of the 8th BSNLP Workshop on Balto-Slavic Natural Language Processing, pages 86–92

highlight ELMo embeddings (Peters et al., 2018), which are trained using a bidirectional language model, the Flair embeddings (Akbik et al., 2018), trained with a character-level language model, and finally embeddings retrieved from Transformer-based models (Vaswani et al., 2017), such as BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019). Any combination of these embeddings can be used as input to a model that is able to predict entity classes, such as a LSTM-CRF (Straková et al., 2019), a biaffine-classifier (Yu et al., 2020), or a linear layer (Devlin et al., 2019).

Particularly relevant to this work is the possibility of using pre-trained cross-lingual embeddings, such as multilingual BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020), both covering a high-number of languages with different scripts. Such models have been shown to perform well on different tasks and languages (Pires et al., 2019; Lewis et al., 2020; Hu et al., 2020).

In the previous edition of the SlavNER shared task (Piskorski et al., 2019) multiple submissions used multilingual BERT to retrieve cross-lingual representations. In particular, Tsygankova et al. (2019) used both word- and character-embeddings as input to a biLSTM-CRF, Arkhipov et al. (2019) further pretrained multilingual BERT on the four target Slavic languages of last shared task's edition and combined its representations with a word-level CRF, and the submission by IIUWR.PL used a combination of different embeddings, where BERT and Flair were included.

3 Approach

Our approach makes use of three key components: a multilingual contextual embedding model, a character-level embedding model, and a biaffine classifier model.

In terms of multilingual contextual embedding models we have explored three options:

1. Multilingual BERT model, which covers 104 languages, and follows the configuration of the BERT-base model (Devlin et al., 2019).

2. XLM-RoBERTa (XLM-R) model (Conneau et al., 2020), trained on 100 languages. We use the large version of the model.

3. Slavic BERT model (Arkhipov et al., 2019), which corresponds to the Multilingual BERT-model further finetuned using resources for Bulgarian, Czech, Polish and Russian. Furthermore, it also rebuilds the original vocabulary to better match these languages.

Besides finetuning the the top-layers of the contextual embedding model during training, we complement these representations with character-level embeddings, obtained with a single-layer CNN.

The biaffine classifier model follows the work by Yu et al. (2020). In particular, the token- and character-level embeddings are concatenated and fed into a Highway BiLSTM (Zhang et al., 2016) which yields a representation for each token. These representations are given to two individual Feed-Forward Neural Networks (FFNNs), responsible for creating a representation that models whether a token is the start/end of a span. Finally, these are passed to a biaffine model, which returns a *scores tensor* with all possible start-end combinations with shape $n \times n \times c$, where n is the number of tokens and c is the number of NER classes plus one, corresponding to the no-entity prediction. This scores tensor masks non-valid spans, i.e., spans where the end position is lower than the start position.

A series of heuristics is then applied to the scores tensor in order to predict spans. First, all the valid spans are retrieved and matched with the corresponding highest-scoring label. Then, all spans whose highest-scoring label corresponds to an entity are sorted by score, from highest to lowest, and are evaluated sequentially. All predicted spans are kept, unless they clash with some of the spans already validated for that input, i.e., unless they overlap with entities that were given an higher score. One of the advantages of this model is that it can model both flat and nested entities, based upon the heuristics we apply.

The model is optimized with the softmax cross-entropy loss.

4 Experimental Setup

4.1 Task

The 3rd edition of the SlavNER Shared Task includes three subtasks: Named Entity Recognition, lemmatization, and cross-lingual entity linking. The available data for this edition adds two extra languages (Slovene and Ukrainian) to the four languages covered in the 2nd edition of the shared task (Bulgarian, Czech, Polish, and Russian).

Our work targets exclusively the subtask of NER, for which there are five types of entities: per-

sons (PER), locations (LOC), organizations (ORG), events (EVT), and products (PRO).

The evaluation of this subtask is case-insensitive, and since the goal is to correctly identify a "*bag-of-mentions*" in a document, it uses three specific metrics, two "relaxed" and one "strict":

- *Relaxed Partial Matching* (RPM) and *Relaxed Exact Matching* (REM), where the system only needs to identify at least one of the forms of a given entity to count a match (e.g. it would only need to identify *Alexandr Kogan* for both *Alexandr Kogan* and *Alexandra Kogana* to be matched). The difference between partial and strict is that the former requires matching only a part of the named entity, while the latter requires an exact match (e.g. in the previous example, matching *Kogan* would be enough for the partial metric).

- *Strict Matching* (SM), where the system has to identify each unique form of a named entity present in a given document (i.e. in the previous example both *Alexandr Kogan* and *Alexandra Kogana* would have to be predicted).

4.2 Data

The train data for the 3rd edition of the SlavNER shared task (SLAVNER2021) includes four topics and covers a total of six languages. The four topics: ASIA_BIBI, BREXIT, NORD_STREAM, and RYANAIR, were part of the data used in the 2nd edition of the SlavNER shared task (SLAVNER2019), apart from minor revisions and two extra languages, Slovene and Ukrainian. Furthermore, there is also an additional generic topic this year, OTHER, which includes only Slovene data. The two added languages have the smallest amount of documents available, 279 and 159 respectively for Slovene and Ukrainian. The other languages have more data available, ranging from 571 in the case of Russian to 918 in the case of Bulgarian.

This edition's test data includes two topics, *"Covid-19"* and *"US_election_2020"*, which are particularly challenging due to the their very specific vocabulary. The most represented language is Slovene, with 333 documents, and the least represented language is Ukrainian, with 168 documents.

Similarly to Tsygankova et al. (2019), we consider data from the 1st edition of the SlavNER shared task, which we will refer to as SLAVNER2017. It includes two topics, EU and TRUMP, and covers seven languages: Czech, Croatian, Polish, Russian, Slovak, Slovene, and Ukrainian. Despite the different set of tags, the extra data can improve the overall performance (Tsygankova et al., 2019).

We use an internal tokenizer that is able to split sentences and also words from punctuation. The data was processed in order to match the format expected by our internal framework used to train NER models, on a sentence-by-sentence basis. This requires matching the document-level annotations, as provided by the organizers of the shared task, with all the individual occurrences in the text. As mentioned in Tsygankova et al. (2019), this leads to two possible errors: matching occurrences of words that do not correspond to entities, and the opposite. The relative difference between the expected number of entities at the document-level and our number of annotations is between 0.78% and 1.6%. Besides the two aforementioned errors, we found the most common mismatches to be related with typos in entities, encoding errors of Latin-text annotations in Cyrillic documents, and errors due to our tokenization.

To make our predictions match the expected format we keep only the unique (case-insensitive) predicted entities, and remove the ones tagged as MISC, obtained when using SLAVNER2017 data.

Unless otherwise noted, train and development data use the same topics and are split by using the top 5% of sentences as development data and the remainder 95% as training data. This split is performed at the level of each topic + language, so that the original ratio of data is kept. SLAVNER2017 data is not included by default.

4.3 Training

We implement our approach using PyTorch (Paszke et al., 2019). The contextual embedding models use the Hugging-Face Transformers library (Wolf et al., 2019), and our biaffine classifier implementation mostly follows the original one[1]. Further training details can be seen in Appendix A.

5 Results

5.1 Preliminary Experiments

Our first set of experiments was conducted using SLAVNER2019 and SLAVNER2021 data. We keep NORD_STREAM and RYANAIR as test topics, and the remainder as train topics. These

[1] https://github.com/juntaoy/biaffine-ner

		F1 - All			F1 - SM - All					
	Model	**RPM**	**REM**	**SM**	**CS**	**RU**	**BG**	**PL**	**UK**	**SL**
2nd ST	Best Reported Scores	90.94	86.40	85.66	84.07	**88.52**	88.25	82.03	-	-
	Multilingual BERT	92.26	88.06	88.62	91.56	84.91	84.12	91.62	-	-
	Slavic BERT	**95.07**	**91.73**	**91.75**	**94.01**	87.14	**91.69**	**93.52**	-	-
	XLM-R Large	94.91	90.68	91.07	93.98	87.11	89.15	92.83	-	-
3rd ST	Multilingual BERT	93.32	88.69	89.32	92.68	85.00	88.28	91.47	**85.05**	89.99
	Slavic BERT	93.54	88.86	89.48	93.38	**87.14**	**92.51**	**93.04**	77.91	85.34
	XLM-R Large	**94.04**	**89.73**	**90.18**	**93.88**	86.14	89.33	92.76	82.86	**90.64**

Table 1: Results obtained for the topics NORD_STREAM and RYANAIR using SLAVNER2019 data (**2nd ST**) and SLAVNER2021 data (**3rd ST**).

	F1 - All - Score Diff		
Base	**RPM**	**REM**	**SM**
XLM-RoBERTa-Large	94.04	89.73	90.18
- Finetune Layers	-1.10	-0.94	-1.21
- Both Topics	-2.97	-3.67	-3.38
- Char Embeds	+0.08	+0.13	+0.18
+ 2017 Data	-0.31	-0.34	-0.31

Table 2: Difference in performance obtained for the NORD_STREAM and RYANAIR topics of the SLAVNER2021 when modifying our approach.

experiments have the following goals: (i) compare our approach's performance with the official SLAVNER2019 scores; and (ii) evaluate the impact of the added languages for the same topics in SLAVNER2021.

For the SLAVNER2019 experiments we used the original test set data. As for the train data, we noticed a mismatch between the number of documents in the original data available for download and the information reported in Piskorski et al. (2019). Since the equivalent data (i.e., the same topics and languages) in this year's data matched both the expected number of documents and entities, we used it instead. For the SLAVNER2021 experiments we use the available data with the aforementioned topic splits.

The obtained results can be seen in Table 1. The impact of further finetuning Multilingual BERT with the four SLAVNER2019 languages in Slavic BERT is noticeable and it results in the best overall scores for that edition's data. However, the extra finetuning step degrades considerably the performance for the two added language in SLAVNER2021, where the model performs worse than any other. This finetune mismatch might partially explain the fact why XLM-R outperforms Slavic BERT in the overall metrics of SLAVNER2021, as opposed to what is observed for SLAVNER2019. Another key aspect for XLM-R's performance is the fact this contextual embedding model is much larger than its BERT-base counterparts (300M vs 120M parameters).

Overall, all models trained with SLAVNER2019 largely outperformed the best scores reported for the 2nd edition of the shared task[2]. We hypothesize these differences are due to: (i) a larger multilingual contextual embedding model, such as XLM-R Large; (ii) a biaffine approach which has been shown to outperform CRF approaches (Yu et al., 2020); (iii) finetuning the top layers of the contextual embedding model during training, which has a positive impact over simple feature extraction (Sun et al., 2019); and (iv) train/development sets using all non-test available topics.

The first hypothesis matches what we have observed in the results presented in Table 1. In particular, the increased model capacity of XLM-R helps to mitigate the multilingual language model tradeoff highlighted by Conneau et al. (2020), "*for a fixed-size model, the per-language capacity decreases as we increase the number of languages*".

The last two hypotheses can be discussed together with the results presented in Table 2, where we perform a simple ablation study. We can observe that not finetuning the top-layers of the contextual embedding model hurts performance. It is plausible to attribute this to the unique characteristics of the languages and entities of the task at hand, where making part of the parameters trainable allows the model to learn better contextual representations for the NER task.

Using both topics as train and development data has a large impact in terms of performance, when compared with using BREXIT as training data and ASIA_BIBI as development data. Even though the scores obtained for the development data are artificially larger, it appears that the model's ability to generalize to new topics is not affected.

The two last results of Table 2 provide us interesting insights. First, it is noticeable that adding SLAVNER2017 data degrades performance. This was something we observed for all SLAVNER2021

[2] http://bsnlp.cs.helsinki.fi/bsnlp-2019/final_ranking.pdf

				F1 - All			F1 - SM - All					
Model		CE	2017	RPM	REM	SM	BG	CS	PL	RU	SL	UK
(S1)	XLM-R	-	-	85.24	79.51	78.92	78.94	**82.10**	84.83	73.48	83.80	76.84
(S2)	XLM-R	x	-	**85.66**	**80.07**	**79.34**	79.94	81.43	**85.38**	**73.65**	**84.24**	**77.95**
(S3)	XLM-R	-	x	84.78	79.51	78.83	**80.00**	80.69	84.76	73.39	83.46	76.27
(S4)	XLM-R	x	x	83.77	78.03	77.82	77.61	80.03	82.92	72.90	82.35	76.67
(S5)	XLM-R/Slavic BERT	-/-	x/-	84.29	78.51	77.99	77.67	80.98	82.94	72.57	83.46	76.27

Table 3: Results obtained for the test set of the 3rd edition of the shared task. **CE** - Includes contextual embeddings. **2017** - Includes SLAVNER2017 data.

	F1 - SM - All				
Model	PER	LOC	ORG	PRO	EVT
S1	**90.88**	90.30	70.17	52.00	**11.63**
S2	90.69	90.67	**70.68**	**54.27**	10.46
S3	90.69	**90.79**	69.74	43.35	05.13
S4	90.55	90.37	69.06	39.62	07.28
S5	89.55	89.75	69.16	50.40	06.11

Table 4: Entity-level results for the test set of the 3rd edition of the shared task.

experiments, and for all SLAVNER2019 experiments excluding the one that used Multilingual BERT. Secondly, removing the character-level embeddings seems to have an almost negligible impact. However, we noticed some variance in scores among runs with regard to this change.

5.2 3rd SlavNER Shared Task Submissions

We have made a total of five submissions to this year's shared task, as detailed in Appendix B. Following the scores reported in Table 1, and the corresponding discussion in Subsection 5.1, we have decided to use XLM-R in most of our submissions. The first four submissions correspond to a XLM-R with all the possible combinations with/without character embeddings and including/not including the SLAVNER2017 data in the training data. We selected these combinations due to the observed variance of scores when using character embeddings, and due to the fact that both Ukrainian and Slovene are part of the SLAVNER2017 data. The fifth submission is an hybrid approach, where the Ukrainian and Slovene documents are predicted with a model trained with a XLM-R using both SLAVNER2021 and SLAVNER2017 data, and the predictions for the remaining languages were obtained with a Slavic BERT model.

Our submissions' scores can be seen in Table 3. The second submission, which uses a XLM-R with character embeddings, scored the highest in terms of F1 for the overall metrics and in four of the six languages in terms of strict matching F1. It is noticeable that adding the SLAVNER2017 data to the training data had a negative impact, and that the impact of character embeddings is variable. The hybrid approach (S5) did not yield the expected scores, since using Slavic BERT as the contextual embedding model did not improve performance for BG/CS/PL/RU, as opposed to what we observed in our preliminary experiments.

With regard to the entity-level results, as observed in Table 4, our scores for PRO and EVT seem subpar. After analyzing the error-log files, we noticed some common mistakes: (i) We miss some quotation marks, e.g., we predict *abcd* instead of *«abcd»*; (ii) Covid-19 related tags are mostly erroneously classified as PRO and not as EVT; and (iii) we miss topic-specific vocabulary. Some of the most common wrong/missed target predictions for the *"covid-19"* topic are *EVT-Covid-19*, *EVT-Pandemic*, and *ORG-BioNTech*, and for the *"us_election_2020"* topic are *PRO-CNN*, *ORG-The-White-House*, and *ORG-Republican-Party-USA*.

At the time of writing this document we do not yet have access to the overall results, and therefore cannot comment on our relative performance.

6 Conclusion

We have proposed a multilingual approach to the 3rd SlavNER Shared Task, where we can make use of a single model trained with multiple source languages to predict NER tags. In particular, we have shown that using a large model, such as XLM-R, coupled with character-level embeddings and a biaffine classifier is able to perform well when replicating the scenario of the 2nd Shared Task on SlavNER, as well as for the languages added to this year's edition of the SlavNER shared task.

Acknowledgments

This work is supported by the EU H2020 SELMA project (grant agreement No 957017) and the Lisbon Regional Operational Programme (Lisboa 2020), under the Portugal 2020 Partnership Agreement, through the European Regional Development Fund (ERDF), within project TRAINER (Nº045347).

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649.

Mikhail Arkhipov, Maria Trofimova, Yurii Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. Mlqa: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 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)*, pages 2227–2237.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

Jakub Piskorski, Laska Laskova, Michał Marcińczuk, Lidia Pivovarova, Pavel Přibáň, Josef Steinberger, and Roman Yangarber. 2019. The second cross-lingual challenge on recognition, normalization, classification, and linking of named entities across slavic languages. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 63–74.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested ner through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Tatiana Tsygankova, Stephen Mayhew, and Dan Roth. 2019. Bsnlp2019 shared task submission: Multi-source neural ner transfer. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 75–82.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476.

Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass. 2016. Highway long short-term memory rnns for distant speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5755–5759. IEEE.

A Training Details

We implement our approach using PyTorch (Paszke et al., 2019). We train models for 60 epochs, evaluating the model twice per epoch, and stop training early if NERC F1 does not improve in the development set after 24 validation steps. All models are optimized with Adam (Kingma and Ba, 2015), with a batch size of 32 and a maximum grad norm of 5. We keep the model with the highest NERC F1 score in the development set. Using scores from 33 experiments we have calculated Pearson Correlation Coefficient values of 0.89/0.79/0.84 between the NERC F1 test set values and the official RPM/REM/SM metrics. Despite the mismatch, the correlation values show that NERC F1 is a good approximation of the official metrics.

The character embeddings are learned during training, and the contextual embedding model, implemented using Hugging-Face Transformers library (Wolf et al., 2019), is further finetuned. In particular, until otherwise mentioned, we freeze all parameters of the contextual embedding model apart from the embedding-layer and the top-4 layers. Moreover, an embedding pooler learns a weighted average of the contextual embedding model's top-4 layers for each token. We have not observed gains from either finetuning more layers, nor using more layers when pooling the representation for a given token. Following Devlin et al. (2019) we represent each token by its first subtoken. Both embeddings models use a learning rate of 5e-5, with a linear scheduler where the maximum value occurs after 10% of the training steps.

All the contextual embedding models are available in the HuggingFace Transformers library, under the names BERT-BASE-MULTILINGUAL-CASED (Multilingual BERT), XLM-ROBERTA-LARGE (XLM-R Large), and DEEPPAVLOV/BERT-BASE-BG-CS-PL-RU-CASED (Slavic BERT). Both multilingual BERT and XLM-R cover all the six languages that are part of this shared task.

Our implementation of the biaffine classifier model mostly follows the original implementation[3]. It uses a learning rate of 1e-3 with a exponential scheduler, implemented from its TensorFlow version[4]. We follow Yu et al. (2020) choice of hyperparameters, as described in Table 5.

Hyperparameter	Value
Char Embeddings Dimension	8
Char Embeddings Filter Size	50
Char Embeddings Filter Width	3,4,5
Embeddings Dropout	0.5
BiLSTM Hidden Dimension	200
BiLSTM Number of Layers	3
BiLSTM Dropout	0.4
FFNN Hidden Dimension	150
FFNN Dropout	0.2
Scheduler Decay Step	100
Scheduler Decay Rate	0.999
Scheduler Staircase	True

Table 5: Classifier hyperparameters.

B Submissions Details

Submission	Details
Submission 1	XLM-R w/o CE w/o 2017
Submission 2	XLM-R w/ CE w/o 2017
Submission 3	XLM-R w/o CE w/ 2017
Submission 4	XLM-R w/ CE w/ 2017
Submission 5	UK/SL: XLM-R w/o CE w/ 2017 Other: Slavic BERT w/o CE w/o 2017

Table 6: Submission details including or not Char Embeddings (CE) and SLAVNER2017 data (2017).

[3] https://github.com/juntaoy/biaffine-ner

[4] See tf.keras.optimizers.schedules.ExponentialDecay