Simple ways to improve NER in every language using markup

Luis Adrián Cabrera-Diego[1], Jose G. Moreno[2], and Antoine Doucet[1]

[1] La Rochelle Université, L3i, La Rochelle, 17031, France
luis.cabrera_diego@univ-lr.fr, antoine.doucet@univ-lr.fr
[2] Université Paul Sabatier, IRIT, Toulouse, 31062, France, jose.moreno@irit.fr

Abstract. We explore three different methods for improving Named Entity Recognition (NER) systems based on BERT, each responding to one of three potential issues: the processing of uppercase tokens, the detection of entity boundaries and low generalization. Specifically, we first explore the marking of uppercase tokens for providing extra casing information. We then randomly mask tokens, as in a masked language model, and predict them along with the NER task to improve NER generalization. Finally, we predict entity boundaries to ameliorate named entity detection. The experiments were done over five languages, three of which are low-resourced: Slovene, Croatian, Finnish, English and Spanish. Results show that predicting masked tokens can be beneficial for most languages, while marking uppercase tokens can be a simple method for dealing with uppercase sentences in NER. Furthermore, our methods improved the state of the art for Croatian and Finnish.

Keywords: Named Entity Recognition · BERT · multi-task

1 Introduction

Named Entity Recognition (NER) is a fundamental task in the processing of texts that consists of extracting entities that semantically refer to notions such as locations, people and organizations [19,32]. In 2019, Devlin et al. [8] presented the deep neural network model called *Bidirectional Encoder Representations from Transformers (BERT)* and demonstrated that pre-trained models based on BERT can be fine-tuned to achieve high performance in multiple tasks. As a consequence, multiple BERT-based NER systems have been created in the last couple of years [12].In this work[1], we explore three different aspects that might play a role in the performance of NER systems:

Uppercase words: Although it is uncommon to have to have texts not following standard casing rules, some NER datasets, such as CoNLL 2003 [26] and CoNLL 2002 [25] may contain a small percentage of sentences with uppercase

[1] github.com/EMBEDDIA/NER_BERT_Multitask

words. These sentences might be harder to predict by systems based on language models where BPE tokenizer are used, such as BERT or RoBERTa [16], because uppercase versions of a word are not tokenized in the same way as their title or lowercase versions [22]. For instance, the words *Italy* and *ITALY*, are split by $\text{BERT}_{\text{BASE}}$ tokenizer as [Italy] and [IT,##AL,##Y], respectively. If we ask $\text{BERT}_{\text{BASE}}$ to predict the masked word in "I live in Rome, [MASK].", the top prediction is *Italy* (50.5%), followed by *too* (8.9%), *though* (3.6%), *Rome* (3.0%) and *now* (2.4%). Nonetheless, predicting the masked word for the same phrase but using uppercase words, i.e. "I LIVE IN ITALY, [MASK].", BERT proposes the following words *too* (4.0%), *please* (1.6%), *then* (1.1%), *now* (1.1%) and *Mom* (0.7%). Moreover, if we mask only one subtoken of the word *ITALY*, BERT produces top predictions such as *##E* (20.7%) *IS* (18.6%) and *AND* (15.7%). The reason that uppercase words are harder to process correctly is that different BPE tokens have different dense representations and, in consequence, the language model might not have enough knowledge about them [24,22]. Therefore, it might be necessary to process differently uppercase words in NER systems.

Entity boundaries: Although the prediction of named entity boundaries is associated to nested named entities [6], in Li et al. [13], the authors determined that the prediction of boundaries in flat named entities in English (CoNLL 2003 [26]) can be as high as 97.40. Therefore, we ask ourselves whether this performance is always reached in all languages and datasets. Or whether, in some cases, the correct prediction of boundaries is a bottleneck for improving the detection of named entities.

Low generalization: One of the biggest challenges in NER systems is the prediction of named entities that were never seen during the training or that have weak or zero regularity, such as titles of books and movies [15]. In the last year, there have been some interesting methods for increasing NER systems' generalization, such as the manual creation of triggers [14] and, the permutation of named entities along with the reduction of context as in Lin et al. [15]. In this work, we explore another method that might improve the generalization while, at the same time, might adequate the language model, to the domain of the dataset analyzed.

We counter these cases with three different approaches that could be used to improve the performance of a NER system. First, we explore whether the marking of uppercase tokens and the addition of supplementary cases can improve the detection of named entities. Second, we determine whether training, in a multi-task fashion, a named entities boundaries detector could improve the performance of a named entity system. Finally, we investigate whether the masking and prediction of tokens during training, could increase the NER system generalization.

Therefore, we present our experiments and conclusions on five different datatsets. Two of them in high-resourced languages: English (CoNLL 2003 [26]) and Spanish (CoNLL 2002 [25]). And three from low-resourced languages: Croatian (HR500k [18]), Slovene (SSJ500k [11]) and Finnish [19]. The obtained results show an improvement over the state of the art for Croatian, but also for Finnish,

while we have interesting results for the rest of the languages. Notably, we can observe a benefit of marking uppercase tokens but also on predicting masked tokens during the training of the models.

The rest of the paper is structured as follows. In Section 2, we present the most relevant related work regarding NER systems for the languages explored in this paper. Then, in Section 3, we introduce the methodology explored in this work. The explored datasets and the experimental setup are described in Section 4 and Section 5, respectively. The results and their discussion are presented in Section 6. Finally, the conclusions and future work are presented in Section 7.

2 Related Work

Recent multilingual NER systems have opted for BERT-based architectures. For instance, Luoma et al. [19] presented a new dataset in Finnish based on the Universal Dependency Finnish corpus and evaluated it using different NER systems from the state of the art, including FinBERT [28], a Finnish BERT.

For Croatian and Slovene, the Janes Project [10] proposed Janes-NER, an NER system that uses a Conditional Random Fields (CRF) classifier, along with lexica and Brown clusters; it is based on the work of Ljubešić and Erjavec [17]. It was trained and tested on HR500k [18] and SSJ500k [11] using 5 possible entity types: Location, Person, Person-Derived, Organization and Miscellaneous. Both languages have been evaluated[2] using the Babushka-Bench[3].

The work of Ulčar and Robnik-Šikonja [27] presented *CroSloEngual* (CSE), a multilingual BERT for Croatian, Slovene and English. The pre-trained model was evaluated on NER using the datasets of HR500k [18] and SSJ500k [11]; only entities of type Location, Person and Organization were predicted.

In Alves et al. [2], the authors evaluated two NER systems from the state of the art: Polyglot [1] and the Croatian NERC System (CNERC) [4] over the corpus HR500k [18]. Only the entities of type Location, Person and Organization were considered.

Yu et al. [32] used BERT [8], FastText [5] and character embeddings, with a biaffine model [9] in a new NER system. Their results improved state-of-the-art results in multiple datasets including Spanish CoNLL 2002 [25].

In Li et al. [13], the authors created *BdryBot* a tool for detecting named entities boundaries. It is based on multiple recursive neural networks, a pointer mechanism and BERT. On English CoNLL 2003 [26], they reached an F-score of *0.974* on the prediction of entity boundaries. Comparing this value with the current state of the art for the detection of named entities, 0.943 [31], it means that the detection of named entity boundaries is easier to achieve than the prediction of their types.

[2] github.com/clarinsi/janes-ner
[3] github.com/clarinsi/babushka-bench

3 Methodology

As indicated previously in Section 1, we present in this work an NER system that along with three methods looks for reducing the effects of three aspects: uppercase words, entity boundaries as bottleneck and low-generalization. The architecture of the proposed methodology is shown in Figure 1 and it is composed of four key elements: prediction of named entities, prediction of entity boundaries, prediction of masked tokens and processing of uppercase tokens. Each of the components is described as follows.

The prediction of named entities is done through a linear layer that it is connected to the output generated by a BERT model, similarly to the work proposed by Devlin et al. [8]. However, to improve the correct annotation of entities, we add as well a CRF such as in Ma and Hovy [20].

For the prediction of named entity boundaries, we make use the same architecture used for the prediction of named entities. However, the linear and CRF layers focus on a reduced set of labels, which are only related to entity boundaries. The objective of this component is to determine whether, the prediction of boundaries could improve the prediction of named entities as the former is an easier task than the latter.

Regarding the prediction of masked tokens, the architecture follows the same one proposed by Devlin et al. [8] for training a masked language model. This consists of introducing the output of a BERT model into a linear layer, which has the same size of the pre-trained vocabulary. The linear layer is expected to predict the masked token. The component's goal is to force BERT to learn patterns that could detect named entities even when a portion of the information is hidden. At the same time, fine-tune BERT embeddings to the domain of the NER dataset.

The prediction components, as show in Figure 1, are coupled in a multi-task fashion. This means, that each of the previously mentioned components, are associated to a specific loss function, which produces values related to each task. During training, the losses produced by all the tasks, are summed. However, during prediction time, as we are only interested in the prediction of named entities, only the NER part is active.

With respect to the tokenization of uppercase words, we decided to follow a pre-processing method. In this case, we explore whether introducing additional cases to the input sentence could bring additional information to BERT regarding the correct context in which a named entity occur. For doing this, we follow an approach similar to the one of Baldini Soares et al. [3], in which entity markers are used to focus BERT on specific information. In the context of uppercase words, we add to BERT's vocabulary two special tokens *[UP], [up]*, which mark the occurrence of an uppercase word. Inside these special tokens, we introduce the tokens produced by the original uppercase word, but also the tokens obtained of the word in title-case and lowercase. For instance, in Figure 1, we can observe that two words are in uppercase, i.e. *ROME* and *ITALY*, thus we change their respective tokens to a marked and enriched version. In other words, the tokens

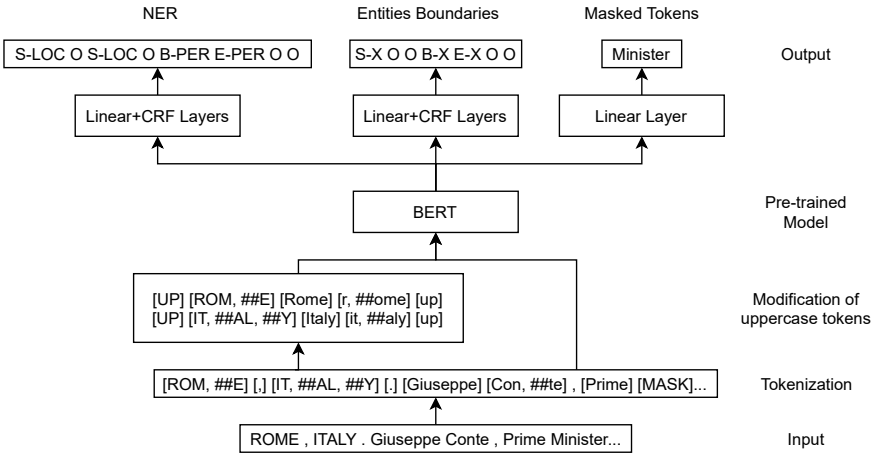

Fig. 1: Proposed architecture, including an example of the expected output.

associated to *ROME* become "[UP] [ROM, ##E] [Rome] [r,##ome] [up]".[4] It should be indicated, that the prediction of named entities (or boundaries), are done uniquely over the first token, which correspond to [UP]. This approach is similar to the one used by Devlin et al. [8] for words split into multiple sub-tokens by BERT's tokenizer but also by Baldani Soares et al. [3], regarding the use of entity markers.

4 Datasets

For English, we use the collection proposed in CoNLL 2003 [26] and for Spanish the dataset created in CoNLL 2002 [25]. Both corpora have been annotated using 4 types of named entities: Location, Person, Organization and Miscellaneous.

For Croatian and Slovene, we use the corpus HR500k [18] and SSJ500k [11], respectively. According to their respective authors, both corpora have been annotated with 5 types of named entities: Location, Person, Person-derived, Organization and Miscellaneous. However, in the case of HR500k, we did not find entries tagged with the Miscellaneous type, as it happened as well in [2,27]. Following some previous works [2,27], we removed the type Person-derived, as it is the less frequent type in both corpora. It should be indicated, that we use the training and testing partitions provided by Ulčar and Robnik-Šikonja [27]. However, the training partitions were split into 90% training and 10% development using a stratified strategy in order to use an early stop approach.

Regarding the Finnish language, we have used the corpus proposed by Luoma et al. [19]. This corpus has been annotated using 6 different types of named entities: Location, Date, Person, Event, Organization and Product.

[4] Neither the square brackets nor the commas surrounding the non-special tokens are in the original representation. However, they are shown in our examples to show the different sub-tokens produced for each word.

5 Experimental Setup

The NER systems explored in this article are based on BERT, using Pytorch, HuggingFace's Transformers [30] and different pre-trained BERT models: for English we make use of $BERT_{BASE}$ [8]; for Finnish, *FinBERT* [28]; for Slovene and Croatian, *CroSloEngual* [27] and for Spanish we use *BETO* [7].

All the named entity tags are encoded using BIOES (Beginning, Inside, Outside/Other, End, Single). For the detection of boundaries, we convert the named entity tags into a generic BIOES encoding; in other words, we use a generic entity type, e.g. B-X, I-X, E-X.

For each language, we train 12 different models. The first model, i.e. baseline, is the implementation that only consists of BERT+Linear+CRF. The remaining 11 models, are the different combinations of the approaches described in Section 3 when added to our baseline. Based on the recommendation proposed by Mosbach et al. [21], every model is trained up to 20 epochs using an early stop approach and AdamW with bias correction along with an epsilon of 1×10^{-8}. The early stop is based on the micro F-score and loss of the development datatset.

With respect to the masking of tokens, we only affect the sentences in the training partitions that are longer than three tokens[5]. At each epoch, we select randomly 25% of each sentence's tokens and substitute them with *[MASK]*. If a token after being processed by BERT's tokenizer produces more than one BERT's token, we randomly select one for masking it. For instance, in the case of the last name *Conte*, see Figure 1, one of the sub-tokens would be masked.

In Table 1, we present a summary of the hyperparameters used for training the NER system. As it can be seen, all the parameters, except for the number of epochs and optimizer, follow those used in Devlin et al. [8].

It should be noted that unlike other works, such as [8,28,19], were BERT's input was enriched either with surrounding sentences or document context, our models have for input only the sentence that needs to be analyzed. Moreover, and in contrast with some BERT implementations, the inputs surpassing BERT's token window size are split instead of truncated.[6] The splitting consists of generating a new input sentence with the rest of the tokens; during prediction, the tokens are aligned to match the original input.

We evaluate the output of the NER system using *Seqeval*[7]. With respect to the assessment of boundaries, we use Nervaluate[8]. This evaluation tool provides *exact*, a metric which determines how well the boundaries of the predicted named entities match those found in the gold standard, regardless of the type.

[5] In this case, we talk about the actual definition of tokens, rather than those obtained by BERT's tokenizer

[6] Some implementations disregard the tokens surpassing the token window or considered these as the type *Other*.

[7] github.com/chakki-works/seqeval

[8] github.com/ivyleavedtoadflax/nervaluate/

Table 1: Hyperparameters used for training the proposed architecture.

Hyperparameter	**Value**
Maximum Epochs	20
Early Stop Patience	3
Learning Rate	2×10^{-5}
Scheduler	Linear with warm-up
Warm-up Ratio	0.1
Optimizer	AdamW with bias correction
AdamW ϵ	1×10^{-8}
Random Seed	12, 24, 58, 89
Dropout rate	0.1
Weight decay	0.01
Clipping gradient norm	1.0
BERT's Sequence Size	128
Masking Percentage	25%
Training Mini-Batch	32
Testing Mini-Batch	8

6 Results and Discussion

We present in Table 2 and Table 3 the average and maximum results of five iterations, in terms of micro and macro F-score, regarding the prediction of named entities for the different combinations of systems proposed in this work. We also present results from the state of the art (only a selection of it for the case of English, where the list could be very long). It should be noted that the scores from the state of the art presented Table 2 and Table 3 are not product of multiple iterations, except for $\text{BERT}_{\text{BASE}}$ [8]. As well, the evaluation of Janes-NER using the Babushka-Bench, see Table 2, does not consider errors in boundaries, and calculates the macro F-score using the performance of 5 named entity types plus the obtained score of predicting the *Other* type. Moreover, for Croatian and Slovene, Table 2, the number of named entity types predicted by each system might not be the same. This last issue will be discussed in detail further ahead.

We can observe, in Table 2 and Table 3, that the prediction of masked tokens can improve, in average, the performance of the explored methods, the exception is Finnish, where the performance decreases. In fact, we can observe that for Finnish, the difference between the maximum macro F-score and the average is larger, meaning that the model becomes less stable when predicting masked tokens. Nonetheless, we can achieve better values of micro F-scores for Finnish. Furthermore, by masking tokens, either with other features or not, we can improve in average the performance in Spanish CoNLL 2002. Nonetheless, our maximum score does not reach the performance presented by Yu et al. [32]. Although for English CoNLL 2003, we are still far from the current state of the art, and slightly worse than $\text{BERT}_{\text{BASE}}$, it should be noted that do not use any kind of document context as Devlin et al. [8] did. This might be a signal that we're forcing BERT to generalize better and to deal with smaller contexts.

It is unclear the reasons of why the masking of tokens affected the stability of the less frequent named entities in Finnish. It could be the case that we did not mask enough tokens to force BERT to generalize in certain iterations. Or it might

Table 2: Average values of micro and macro F-score for each experiment over five iterations on Croatian, Slovene and Finnish. The maximum value of each iteration is between brackets. The best performance is in bold. *Boundaries (B.), Uppercase (U.), Generated Uppercase (G.U.) and Masked (M.)*

	Croatian		**Slovene**		**Finnish**	
	Macro	Micro	Macro	Micro	Macro	Micro
FinBERT [19]	-	-	-	-	81.00	**91.60**
BiLSTM [19]	-	-	-	-	-	81.50
Janes [10]	67.30	-	75.20	-	-	-
CSE [27]	88.40	-	**92.00**	-	-	-
Polyglot [1,2]	62.20	-	-	-	-	-
CNERC [4,2]	65.40	-	-	-	-	-
Baseline	87.35 (88.79)	84.78 (86.32)	85.30 (86.95)	89.93 (90.49)	81.35 (82.03)	91.20 (91.76)
B.	87.09 (87.92)	84.60 (86.08)	84.33 (86.33)	89.45 (90.61)	80.78 (83.05)	90.38 (90.88)
U.	87.64 (88.35)	85.21 (86.90)	85.12 (86.70)	89.39 (89.83)	80.41 (83.25)	90.32 (91.26)
G.U.	88.08 (**89.54**)	**86.07** (**88.12**)	85.86 (86.74)	89.54 (90.57)	**82.41** (85.23)	91.33 (91.82)
B.+U.	87.85 (88.53)	85.91 (86.77)	84.23 (88.06)	88.74 (90.68)	80.20 (82.61)	90.32 (90.73)
B.+G.U.	87.46 (88.78)	85.20 (87.55)	85.83 (86.67)	89.56 (90.09)	82.14 (83.24)	91.19 (91.82)
M.	87.50 (88.04)	85.06 (85.54)	85.50 (86.97)	90.56 (91.42)	80.20 (83.54)	90.81 (91.50)
M.+B.	87.54 (88.21)	85.46 (86.41)	84.21 (86.62)	89.81 (90.68)	79.33 (81.52)	90.86 (91.45)
M.+U.	88.20 (89.36)	85.83 (87.30)	86.32 (87.79)	90.53 (91.10)	79.81 (83.46)	91.05 (**92.09**)
M.+G.U.	88.21 (89.36)	86.03 (88.04)	86.46 (87.21)	90.53 (90.72)	78.53 (81.93)	90.89 (91.14)
M.+B.+U.	88.20 (89.05)	85.90 (86.94)	86.26 (87.89)	90.35 (91.40)	80.23 (82.09)	91.27 (**92.09**)
M.+B.+G.U.	87.70 (89.33)	85.64 (87.70)	87.18 (89.56)	**91.07** (**91.90**)	81.85 (**85.66**)	90.87 (91.64)

be related of the agglutinative characteristic of Finnish, in which the masking of tokens affect severely key elements of the language to predict correctly named entities.

In Table 4, we present the results regarding the *exact* metric in terms of micro and macro F-score for each language. This metric evaluates how well a system predicts the boundaries of named entities regardless of the type associated. For Croatian, English and Spanish, we can observe in Table 4 that the prediction of entity boundaries is quite stable in general, either in terms of micro or macro F-score. It should be indicated, that the state-of-the-art micro F-score for English CoNLL 2003 regarding the detection of boundaries is the following: BdryBot 95.90, BERT 96.90 and BdryBot+BERT 97.40 [13]. Regarding Slovene, we can notice that prediction of masked tokens improve the exact metric, however training a model where we predict along the boundaries seems to do not have any effect in general. For Finnish, the exact metrics shows an stable performance in terms of micro F-score, nonetheless, in terms of macro F-score we can observe a decrement when we predict masked tokens.

Another element to discus regarding Table 4, is that for languages such as English and Spanish, recognizing named entity boundaries can be considered an easy task. Nevertheless, for for Croatian, and in lesser degree Slovene and Finnish, it is much more difficult. Moreover, for Finnish the prediction of entity boundaries is harder to achieve for the less frequent types of named entities as it shows the larger difference between the micro and macro F-scores. Furthermore, the results obtained in Table 4 give us an idea on what could be the maximum possible score that a NER system could achieve if the task would only consist of predicting the types of confirmed named entity boundaries. In the same line, the results show that despite the fact that it is easy to find named entity boundaries

Table 3: Average values of micro and macro F-score for each experiment over five iterations on Spanish and English; the maximum of each iteration is between brackets. The best performance is in bold. *Boundaries (B.), Uppercase (U.), Generated Uppercase (G.U.) and Masked (M.)*

	Spanish		**English**	
	Macro	Micro	Macro	Micro
BERT Base [8]	-	-	-	92.40
LUKE [31]	-	-	-	**94.30**
Seq2seq+BERT [23]	-	88.80	-	*92.90*
NER Dep.Par. [32]	-	**90.30**	-	93.50
Baseline	85.85 (86.55)	88.06 (88.57)	90.21 (90.70)	91.48 (91.82)
B.	86.00 (87.05)	88.20 (88.70)	90.00 (90.19)	91.41 (91.51)
U.	85.69 (87.33)	87.81 (88.71)	89.97 (90.27)	91.51 (91.82)
G.U.	86.53 (86.78)	88.30 (88.82)	90.43 (90.97)	91.89 (92.20)
B.+U.	85.32 (86.24)	87.67 (88.14)	89.91 (90.20)	91.39 (91.62)
B.+G.U.	85.76 (86.47)	87.94 (88.52)	90.60 (90.77)	91.93 (92.05)
M.	87.19 (88.02)	**88.99** (89.40)	90.51 (90.72)	91.86 (92.05)
M.+B.	87.18 (88.24)	88.89 (89.51)	90.32 (90.60)	91.72 (91.93)
M.+U.	**87.27** (**88.38**)	88.83 (**89.56**)	**90.71** (90.96)	92.17 (92.31)
M.+G.U.	86.75 (87.93)	88.47 (89.43)	90.69 (90.89)	92.11 (92.28)
M.+B.+U.	86.55 (87.64)	88.32 (89.18)	90.54 (90.81)	92.02 (92.27)
M.+B.+G.U.	86.62 (87.52)	88.39 (89.22)	90.85 (**91.27**)	92.22 (92.62)

Table 4: Average values of micro and macro F-score for the exact metric, which evaluates the correct prediction of entity boundaries regardless their type, over five iterations. *Boundaries (B.), Uppercase (U.), Generated Uppercase (G.U.) and Masked (M.)*

	Croatian		**Slovene**		**Finnish**		**Spanish**		**English**	
	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro
Baseline	89.81	87.24	90.35	93.72	83.05	92.54	93.19	94.93	94.80	95.67
B.	89.88	87.41	90.52	93.92	83.62	92.27	93.24	94.96	94.66	95.66
U.	90.14	87.73	89.77	93.30	82.37	92.26	93.28	94.97	94.53	95.66
G.U.	90.50	88.56	90.84	93.68	84.43	**92.86**	93.65	**95.09**	94.99	96.00
B.+U.	90.54	**88.62**	89.84	93.50	82.98	92.31	92.73	94.65	94.70	95.76
B.+G.U.	90.04	87.88	92.01	94.54	**84.53**	92.66	93.29	95.02	95.06	95.97
M.	90.12	87.85	90.71	94.22	81.39	91.74	93.41	94.94	94.80	95.71
M.+B.	90.55	88.47	90.49	93.97	81.42	91.92	**93.77**	95.04	94.87	95.77
M.+U.	90.59	88.41	91.47	94.54	81.13	92.11	93.63	95.00	94.83	95.86
M.+G.U.	90.36	88.35	91.24	94.31	80.20	91.96	93.33	94.77	95.06	95.99
M.+B.+U.	**90.58**	88.44	91.68	94.68	81.82	92.33	93.32	94.86	94.87	95.89
M.+B.+G.U.	90.07	88.03	**92.22**	**94.98**	83.06	91.94	93.31	94.85	**95.10**	**96.02**

in Spanish, predicting their type is much more difficult for that language, compared to English or Croatian, if we compare the results shown in Table 2 and Table 3. In Croatian, despite the fact that the detection of entity boundaries is difficult, the prediction of their types lies in a range of around three points, while in Spanish it is around seven points. Therefore, we can deduct, that in order to improve the prediction of named entities in languages such as the Croatian, it is necessary to primarily focus on the correct detection of boundaries. Further, for Spanish, it is necessary to improve the prediction of types rather than entity boundaries. However, this last issue could also be a sign of discrepancies in the annotation, either of the training or the testing dataset, something that is already known to occur in the English CoNLL 2003 corpus [29].

With respect to the marking of uppercase tokens, we can notice in Table 2 and Table 3, that we can improve the performance mainly in English and Croatian.

Table 5: Example of predictions for uppercase sentences in different languages. All sentences, except for English, are partial (the models processed the full sentence) and were converted into uppercase words before processing them. *Boundaries (B.), Uppercase (U.), Generated Uppercase (G.U.) and Masked (M.)*

Sentence	SOCCER	-	LEEDS	'	BOWYER	FINED	FOR	PART	IN	FAST-FOOD	FRACAS	.
Baseline	O	O	S-PER	O	O	O	O	O	O	O	O	O
M.+U.	O	O	S-ORG	O	S-PER	O	O	O	O	O	O	O
M.+B.+G.U.	O	O	S-ORG	O	S-PER	O	O	O	O	O	O	O
Gold-Std.	O	O	S-ORG	O	S-PER	O	O	O	O	O	O	O

(a) English: Soccer - Leeds' Bowyer fined for part in fast-food fracas.

Tokens	KENIAN	ANGLIKAANISEN	KIRKON	SIHTEERI	JOHN	KAGO	SANOI
Baseline	O	O	O	O	B-PER	E-PER	O
M.+U.	O	B-ORG	I-ORG	E-ORG	O	O	O
M.+G.U.	S-LOC	B-ORG	E-ORG	O	B-PER	E-PER	O
Gold-Std.	B-ORG	E-ORG	E-ORG	O	B-PER	E-PER	O

(b) Finnish: John Kago, secretary of the English Church of Kenya, said Thursday

Tokens	PROFESOR	NA	USLA	STEPHEN	HUBBELL
Baseline	O	O	O	B-PER	E-PER
U.	O	O	O	B-PER	E-PER
G.U.	O	O	S-ORG	B-PER	E-PER
Gold-Std.	O	O	S-ORG	B-PER	E-PER

(c) Croatian: Professor of USLA Stepehn Hubbell

Tokens	EL	TRIBUNAL	DE	DEFENSA	DE	LA	COMPETENCIA	DETERMINE
Baseline	O	O	O	O	O	O	O	O
M.+U.	O	S-ORG	O	O	O	O	O	O
G.U.	O	B-ORG	I-ORG	I-ORG	I-ORG	I-ORG	E-ORG	O
Gold-Std.	O	B-ORG	I-ORG	I-ORG	I-ORG	I-ORG	E-ORG	O

(d) Spanish: The Competition Defense Court determines

However, by generating random uppercase sentence during training, the marking of uppercase tokens can improve the performance in all the languages. In most cases, this happens as well when applied with other methods such as prediction of masked tokens, specially in Slovene and English.

Although all the datasets contain a variable number of words only in uppercase, there are two possible reasons why some languages benefited more than others. First, it can be the case that the number of uppercase tokens was not large enough to make BERT learn about the marking. Second, it can be related to the textual information that was used to train each BERT model. Nonetheless, BERT is indeed capable of learning the meaning and context of uppercase tokens if enough data has been used during their training, as we did when we generated artificially uppercase sentences.

In Figure 5, we present four examples regarding the prediction of named entities in uppercase sentences; the selected models are the best of each type. As shown in Figure 5b, the prediction of entities does not become perfect when marking uppercase words, but it can definitely improve their recognition.

Table 6: Best F-score values of the three common named entities for the Croatian and Slovene systems. *Boundaries (B.), Uppercase (U.), Generated Uppercase (G.U.) and Masked (M.)*

	Croatian				**Slovene**			
	PER	LOC	ORG	Macro F1	PER	LOC	ORG	Macro F1
CroSloEngual [27]	NA	NA	NA	88.40	NA	NA	NA	**92.00**
Janes-NER [10]	89.00	85.00	72.00	82.00	89.00	80.00	67.00	78.60
Polyglot [1] [2]	NA	NA	NA	62.20	-	-	-	-
Croatian NERC [4] [2]	NA	NA	NA	64.00	-	-	-	-
Baseline	85.40	97.41	83.55	88.79	96.88	90.78	79.28	88.98
G.U.	85.71	95.80	87.19	**89.54**	95.94	91.98	81.98	89.96
B.+G.U.	85.59	95.20	87.30	89.36	95.22	91.61	79.45	88.76
M.+B.+G.U.	87.47	94.92	85.60	89.33	95.34	92.79	84.68	90.93

As indicated previously, the evaluation of NER systems over the Croatian and Slovene datasets is not standard along the state-of-the-art systems. The main reasons is that some named entity types are either not found in the corpus or are disregarded due to their small frequency. Therefore, we present in Table 6 the recalculation of the macro F-scores. These scores are based on the three common types of named entities used in the different NER systems from the state of the art.

With respect to Croatian, we can observe in Table 2 and Table 6 that we can improve the results with respect to CroSloEngual, which is based as well on BERT. Furthermore, our largest improvement, with respect to Janes-NER [10], is for the prediction of named entities of type Location and Organization. For Slovene, we are not able to surpass the performance showed in [27].

Although it is common in the state of the art to present results for the Slovene and Croatian corpora only in terms of macro F-score, the lack of micro F-scores or detailed results per class makes it difficult to perform a detailed comparison of the systems. First, the Croatian and Slovene corpora are not balanced, in other words, the number of entities for each class is not equal. Therefore, the macro F-scores consider equally important all the types of named entities, but disregard their frequency in the dataset. Thus, it is impossible to know whether systems, such as CroSloEngual, Polyglot or Croatian NERC, are focusing either on the most frequent classes or the less frequent ones. For instance, we know that our Croatian NER system focuses on the less frequent class Location (117 occurrences) rather than on the most frequent ones (Person and Organization, respectively with 228 and 365 occurrences). However, our Slovene NER system focuses on the most frequent classes Person and Location (respectively with 257 and 210 occurrences), rather than on the less frequent ones Organization (112 occurrences) and Miscellaneous (47 occurrences).

Despite not surpassing the current state of the art in Slovene, it should be indicated that we trained a system with four types of entities rather than three as it was done in the work of Ulčar and Robnik-Šikonja [27]. This aspect can introduce noise or increases the difficulty of the task, as the left out named entity

Table 7: Results, in terms of F-score, for each entity type for the Finnish corpus and their associated macro and micro F-score; between brackets we indicate the seed that produces these values. *Boundaries (B.), Uppercase (U.), Generated Uppercase (G.U.) and Masked (M.)*

	PER	**LOC**	**ORG**	**DATE**	**EVT**	**PROD**	**Macro F1**	**Micro F1**
FinBERT [19]	95.20	94.70	90.20	**96.80**	43.50	65.80	81.00	91.60
Baseline (12)	94.29	94.34	89.22	95.69	50.00	68.61	82.03	91.42
Baseline (58)	94.42	**95.96**	88.63	96.55	47.06	69.12	81.95	91.76
M.+B.+G.U. (24)	95.19	92.99	89.05	96.07	**66.67**	73.97	**85.66**	91.64
M.+B.+G.U. (12)	**95.27**	94.16	**90.43**	95.28	42.86	74.13	82.02	**92.09**
M.+U. (89)	95.06	93.70	90.20	96.10	50.00	**75.71**	83.46	**92.09**

type, Miscellaneous, is the least frequent one. In this case, if we compare with Janes-NER [10], this systems gets an F-score for Miscellaneous of *27.00* while our masked baseline reaches at most *85.42*.

Finally, with respect to Finnish, we were able to surpass, in average, the performance of the state of the art in terms of macro F-score, and in some iterations the micro F-score. Based on the difference between micro and macro F-score, presented in Table 2, we can determine that multiple of our systems focused slightly more on the less frequent classes, in comparison to the work of [19]. This can be observed in detail in Table 7, where we improved the prediction of entities of type Event and Product, which are the less frequent classes (7 and 79 instances in the test set respectively), by reducing the correct prediction of a more frequent class, i.e. Organization (208). We can observe as well, that macro F-scores can variate more than micro F-scores values, depending on the seed utilized during training. Despite all, we can see as well, that we can improve the predictions of entities without having to add supplementary sentences for increasing the context as Luoma et al. [19] did.

7 Conclusions and future work

Named Entity Recognition (NER) is a task that aims to extract and classify groups of tokens referring to specific types like locations, persons and organizations. In the last couple of years, with the creation of BERT [8], multiple NER systems made use of its architecture to provide high-performing tools. Nonetheless, we observed that this kind of systems could face some issues, such as the bad prediction of uppercase sentences, the wrong detection of entity boundaries and low generalization.

Therefore, in this work, we presented three different methods that could alleviate these issues. Experiments were done over five languages, three of them low-resourced ones. We improved the state of the art with a micro F-score of up to *89.54* in Croatian by marking uppercase tokens and generating uppercase sentences during training. By marking uppercase tokens, predicting boundaries and tokens, we managed to improve the performance of $BERT_{BASE}$ to an F-

score of up to *92.62* in English, while getting the second-best performance in Spanish with an F-score of up to *89.56*. In Finnish, we improved, in average, the prediction of the less frequent named entity types, with a macro F-score of *82.41* versus *81.00* in the state of the art, while reaching a micro F-score of up to *92.09* versus *91.60*. We could also provide a NER system for Slovene that predicts 4 types of named entities, one of which is infrequent, with results comparable to those of another tool from the state of the art that only predicts the three most frequent types.

Furthermore, we observed that in Croatian, the prediction of named entity boundaries is a bottleneck for the NER systems. While in Spanish, it seems to be easy to find the boundaries of named entities, but much harder to determine their type. Finally, we propose a simple method that could improve the prediction of named entities in sentences that are in uppercase words.

In the future, we intend to experiment with additional languages. We would like to asses whether the addition of some context to the left of the split sentences could improve the performance of the NER.

Acknowledgments

This work was supported by the European Union's Horizon 2020 research and innovation program under grants 770299 (NewsEye) and 825153 (Embeddia).

References

1. Al-Rfou, R., Kulkarni, V., Perozzi, B., Skiena, S.: POLYGLOT-NER: Massive Multilingual Named Entity Recognition. CoRR **abs/1410.3791** (2014), `http://arxiv.org/abs/1410.3791`
2. Alves, D., Thakkar, G., Tadić, M.: Evaluating Language Tools for Fifteen EU-official Under-resourced Languages. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 1866–1873. European Language Resources Association, Marseille, France (May 2020)
3. Baldini Soares, L., FitzGerald, N., Ling, J., Kwiatkowski, T.: Matching the Blanks: Distributional Similarity for Relation Learning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2895–2905. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1279
4. Bekavac, B., Tadić, M.: Implementation of Croatian NERC System. In: Proceedings of the Workshop on Balto-Slavonic Natural Language Processing. pp. 11–18. Association for Computational Linguistics, Prague, Czech Republic (Jun 2007), `https://www.aclweb.org/anthology/W07-1702`
5. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. Transactions of the Association of Computational Linguistics **5**, 135–146 (2017)
6. Cao, J., Wang, G., Li, C., Ren, H., Cai, Y., Wong, R.C.W., Li, Q.: Incorporating Boundary and Category Feature for Nested Named Entity Recognition. In: Nah, Y., Cui, B., Lee, S.W., Yu, J.X., Moon, Y.S., Whang, S.E. (eds.) Database Systems for Advanced Applications. pp. 209–226. Springer International Publishing, Cham (2020)

7. Cañete, J., Chaperon, G., Fuentes, R., Pérez, J.: Spanish Pre-Trained BERT Model and Evaluation Data. In: PML4DC at ICLR 2020 (2020)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423
9. Dozat, T., Manning, C.D.: Deep Biaffine Attention for Neural Dependency Parsing. CoRR **abs/1611.01734** (2016), http://arxiv.org/abs/1611.01734
10. Fišer, D., Ljubešić, N., Erjavec, T.: The Janes project: language resources and tools for Slovene user generated content. Language Resources and Evaluation **54**(1), 223–246 (Mar 2020). https://doi.org/10.1007/s10579-018-9425-z
11. Krek, S., Dobrovoljc, K., Erjavec, T., Može, S., Ledinek, N., Holz, N., Zupan, K., Gantar, P., Kuzman, T., Čibej, J., Arhar Holdt, Š., Kavčič, T., Škrjanec, I., Marko, D., Jezeršek, L., Zajc, A.: Training corpus ssj500k 2.2 (2019), http://hdl.handle.net/11356/1210, Slovenian language resource repository CLARIN.SI
12. Li, J., Sun, A., Han, J., Li, C.: A Survey on Deep Learning for Named Entity Recognition. IEEE Transactions on Knowledge and Data Engineering pp. 1–1 (2020)
13. Li, J., Sun, A., Ma, Y.: Neural Named Entity Boundary Detection. IEEE Transactions on Knowledge and Data Engineering pp. 1–1 (2020)
14. Lin, B.Y., Lee, D.H., Shen, M., Moreno, R., Huang, X., Shiralkar, P., Ren, X.: TriggerNER: Learning with Entity Triggers as Explanations for Named Entity Recognition. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 8503–8511. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.752
15. Lin, H., Lu, Y., Tang, J., Han, X., Sun, L., Wei, Z., Yuan, N.J.: A Rigorous Study on Named Entity Recognition: Can Fine-tuning Pretrained Model Lead to the Promised Land? In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 7291–7300. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.592
16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019)
17. Ljubešić, N., Erjavec, T.: Corpus vs. Lexicon Supervision in Morphosyntactic Tagging: the Case of Slovene. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 1527–1531. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016), https://www.aclweb.org/anthology/L16-1242
18. Ljubešić, N., Klubička, F., Agić, Ž., Jazbec, I.P.: New Inflectional Lexicons and Training Corpora for Improved Morphosyntactic Annotation of Croatian and Serbian. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 4264–4270. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016), https://www.aclweb.org/anthology/L16-1676
19. Luoma, J., Oinonen, M., Pyykönen, M., Laippala, V., Pyysalo, S.: A Broad-coverage Corpus for Finnish Named Entity Recognition. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 4615–4624. European Language Resources Association, Marseille, France (May 2020), https://www.aclweb.org/anthology/2020.lrec-1.567

20. Ma, X., Hovy, E.: 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). pp. 1064–1074. Association for Computational Linguistics, Berlin, Germany (Aug 2016). https://doi.org/10.18653/v1/P16-1101
21. Mosbach, M., Andriushchenko, M., Klakow, D.: On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines (2020), https://arxiv.org/abs/2006.04884
22. Powalski, R., Stanislawek, T.: UniCase – Rethinking Casing in Language Models (2020), arXiv cs.CL eprint: 2010.11936
23. Straková, J., Straka, M., Hajic, J.: Neural Architectures for Nested NER through Linearization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 5326–5331. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1527
24. Sun, L., Hashimoto, K., Yin, W., Asai, A., Li, J., Yu, P., Xiong, C.: Adv-BERT: BERT is not robust on misspellings! Generating nature adversarial samples on BERT. arXiv preprint arXiv:2003.04985 (2020)
25. Tjong Kim Sang, E.F.: Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In: COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002) (2002), https://www.aclweb.org/anthology/W02-2024
26. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. pp. 142–147 (2003), https://www.aclweb.org/anthology/W03-0419
27. Ulčar, M., Robnik-Šikonja, M.: FinEst BERT and CroSloEngual BERT. In: Sojka, P., Kopeček, I., Pala, K., Horák, A. (eds.) Text, Speech, and Dialogue. pp. 104–111. Springer International Publishing, Cham (2020)
28. Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., Ginter, F., Pyysalo, S.: Multilingual is not enough: BERT for Finnish (2019), https://arxiv.org/abs/1912.07076
29. Wang, Z., Shang, J., Liu, L., Lu, L., Liu, J., Han, J.: CrossWeigh: Training Named Entity Tagger from Imperfect Annotations. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 5154–5163. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1519
30. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., Platen, P.v., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: HuggingFace's Transformers: State-of-the-art Natural Language Processing. ArXiv **abs/1910.03771** (2019)
31. Yamada, I., Asai, A., Shindo, H., Takeda, H., Matsumoto, Y.: LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6442–6454. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.523
32. Yu, J., Bohnet, B., Poesio, M.: Named Entity Recognition as Dependency Parsing. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 6470–6476. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.577