Exploring Cross-sentence Contexts for Named Entity Recognition with BERT

Jouni Luoma
TurkuNLP group
University of Turku
Turku, Finland
jouni.a.luoma@utu.fi

Sampo Pyysalo
TurkuNLP group
University of Turku
Turku, Finland
sampo.pyysalo@utu.fi

Abstract

Named entity recognition (NER) is frequently addressed as a sequence classification task with each input consisting of one sentence of text. It is nevertheless clear that useful information for NER is often found also elsewhere in text. Recent self-attention models like BERT can both capture long-distance relationships in input and represent inputs consisting of several sentences. This creates opportunities for adding cross-sentence information in natural language processing tasks. This paper presents a systematic study exploring the use of cross-sentence information for NER using BERT models in five languages. We find that adding context as additional sentences to BERT input systematically increases NER performance. Multiple sentences in input samples allows us to study the predictions of the sentences in different contexts. We propose a straightforward method, Contextual Majority Voting (CMV), to combine these different predictions and demonstrate this to further increase NER performance. Evaluation on established datasets, including the CoNLL'02 and CoNLL'03 NER benchmarks, demonstrates that our proposed approach can improve on the state-of-the-art NER results on English, Dutch, and Finnish, achieves the best reported BERT-based results on German, and is on par with other BERT-based approaches in Spanish. We release all methods implemented in this work under open licenses.

1 Introduction

Named entity recognition (NER) approaches have evolved through various methodological phases, broadly including rule/knowledge-based, unsupervised, feature engineering and supervised learning, and feature inferring approaches (Yadav and Bethard, 2018; Li et al., 2020a). The use of cross-sentence information in some form has been a normal part of many NER methods in the former categories, but its role has diminished with the current feature inferring deep learning based approaches. Rule/knowledge-based approaches such as that of Mikheev et al. (1998) typically match strings to lexicons and similar domain knowledge sources, possibly going through text multiple times with refinement based on entities found on earlier passes. Later, manually engineered features were used to incorporate information from the surrounding text, whole documents, data sets and also from external sources. The number of different features and classifiers grew during the years and it was normal that the features also contained cross-sentence information in some form as for example in (Krishnan and Manning, 2006). Dense representations of text such as word, character, string and subword embeddings first started to appear in NER methods as additional features given to classifiers (Collobert et al., 2011). Step by step, feature engineering has been demoted to a lesser role, as the most recent deep learning approaches learn to create meaningful and context-sensitive representations of text by pre-training with vast amounts of unlabelled data. These contextual representations are often used directly as features for existing NER architectures or fine-tuned with labelled data to match a certain task.

In recent years, the development of Natural Language Processing (NLP) in general and NER in particular have been greatly influenced by deep transfer learning methods capable of creating contextual

Proceedings of the 28th International Conference on Computational Linguistics, pages 904–914
Barcelona, Spain (Online), December 8-13, 2020

representations of text, to the extent that many of the state-of-the-art NER systems mainly differ from one another on the basis of how these contextual representations are created (Peters et al., 2018; Devlin et al., 2018; Akbik et al., 2018; Baevski et al., 2019). Using such models, sequence tagging tasks are often approached one sentence at a time, essentially discarding any information available in the broader surrounding context, and there is only little recent study on the use of cross-sentence context – sentences around the sentence of interest – to improve sequence tagging performance. In this paper, we present a comprehensive exploration of the use of cross-sentence context for named entity recognition, focusing on the recent BERT deep transfer learning models (Devlin et al., 2018) based on self-attention and the transformer architecture (Vaswani et al., 2017). BERT uses a fixed-size window that limits the amount of text that can be input to the model at one time. The model maximum window size, or *maximum sequence length*, is fixed during pre-training, with 512 wordpieces a common choice. This window fits dozens of typical sentences of input at a time, allowing us to include extensive sentence context. Here, we first study the effect of predicting tags for individual sentences when they are moved around the window, surrounded by their original document context from the source data. Second, we utilize different predictions for the same sentences to potentially further improve performance, combining predictions using majority voting, adapting an approach that has been used already in early NER implementations (Tjong Kim Sang et al., 2000; Van Halteren et al., 2001; Florian et al., 2003). We evaluate these approaches on five languages, contrasting NER results using BERT without cross-sentence information, sentences in context, and aggregation using Contextual Majority Voting (CMV) on well-established benchmark datasets. We show that using sentences in context consistently improves NER results on all of the tested languages and CMV further improves the results in most cases. Comparing performance to the current state-of-the-art NER results in the 5 languages, we find that our approach establishes new state-of-the-art results for English, Dutch, and Finnish, the best BERT-based results on German, and effectively matches the performance of a BERT-based method in Spanish.

2 Related work

The state-of-the-art in NER has recently moved from approaches using word/character representations and manually engineered features (Passos et al., 2014; Chiu and Nichols, 2016) toward approaches directly utilizing deep learning-based contextual representations (Akbik et al., 2018; Peters et al., 2018; Devlin et al., 2018; Baevski et al., 2019) while adding few manually engineered features, if any. While successful in terms of NER performance, these approaches have tended to predict tags for one sentence at a time, discarding information from surrounding sentences.

One recent method taking sentence context into account is that of Akbik et al. (2019), which addresses a weakness of an earlier contextual string embedding method (Akbik et al., 2018), specifically the issue of rare word representations occurring in underspecified contexts. Akbik et al. (2019) make the intuitive assumption that such occurrences happen when a named entity is expected to be known to the reader, i.e. the name is either introduced earlier in text or is of general in-domain knowledge. Their approach is to maintain a memory of contextual representations of each unique word/string in text and pool together contextual embeddings of a string occurring in text with the contextual embeddings of the same string earlier in text. This pooled contextual embedding is then concatenated with the current contextual embedding to get the final embedding to use in classification.

Another recent approach taking broader context into account for NER was proposed by Luo et al. (2020), where in addition to token representations, also sentence and document level representations are calculated and used for classification using a CRF model. A sliding window is used by Wu and Dredze (2019) so that part of the input is preserved as context when the window is moved forward in text. Baevski et al. (2019) state that they use longer paragraphs in pre-training their model, but it is not mentioned in the paper if such longer paragraphs are used also in fine-tuning the model or predicting tags for NER. Some other approaches such as that of Liu et al. (2019a) include explicit global information in the form of e.g. gazetteers. Also, some approaches formulate NER as a span finding task instead of sequence labelling (Banerjee et al., 2019; Li et al., 2020b). These approaches would likely allow the use of longer sequences, but the incorporation of cross-sentence information is not explicitly proposed by the authors.

In the paper introducing BERT, Devlin et al. (2018) write in the description of their NER evaluation "we include the maximal document context provided by the data." However, no detailed description of how this inclusion was implemented is provided, and some NER implementations using BERT have struggled to reproduce the results of the paper.[1,2] The addition of document context to NER using BERT is discussed also by Virtanen et al. (2019), who fill each input sample with the following sentences and use the first sentence in each sample for predictions, and thus only introduce context appearing *after* the sentence of interest in the source text.

Of the related work discussed above, our approach most closely resembles that of Virtanen et al. (2019), which in turn aims to directly follow Devlin et al. (2018). By contrast to other studies discussed above, we do not introduce extra features or embeddings representing cross-sentence information or incorporate extra information in addition to that captured by the BERT model. Instead, we directly utilize the BERT architecture and rely on self-attention and voting to combine predictions for sentences in different contexts.

3 Data

The data used in this study consists of pre-trained BERT models and NER datasets for five different languages. We aimed to use monolingual BERT models as numerous recent studies have suggested that well-constructed language-specific models outperform multilingual ones (Virtanen et al., 2019; Vries et al., 2019; Le et al., 2020). We selected the following language-specific pre-trained BERT models for our study, focusing on languages that also have established benchmark data for NER:

- BERTje base, Cased for Dutch (Vries et al., 2019)[3]
- BERT-Large, Cased (Whole Word Masking) for English [4]
- FinBERT base, Cased for Finnish (Virtanen et al., 2019)[5]
- German BERT, Cased for German [6]
- BETO, Cased for Spanish (Cañete et al., 2020)[7] .

For comparison purposes we also tested multilingual BERT[8] with the Spanish language. From the models introduced above all except German and multilingual BERT have used the Whole Word Masking variation of the Masked Language Model objective in pre-training instead of the method introduced in the original paper (Devlin et al., 2018). Whole Word Masking was introduced by the developers of BERT after the original paper was published. In this pre-training objective, all the tokens corresponding to one word in text are masked instead of completely random tokens, which often leaves some of the tokens in multi-token words unmasked. We aimed to apply sufficiently large, widely-used benchmark datasets for evaluating NER results, assessing our methods primarily on the CoNLL'02 and CoNLL'03 Shared task Named entity recognition datasets (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003), which cover four of our five target languages. For the fifth language, Finnish, we use two recently published named entity recognition corpora (Ruokolainen et al., 2019; Luoma et al., 2020)[9,10]. These two Finnish datasets are annotated in a compatible way, and for this study they are combined into a single corpus by simple concatenation, following Luoma et al. (2020).

[1] https://github.com/google-research/bert/issues/581
[2] https://github.com/google-research/bert/issues/569
[3] https://github.com/wietsedv/bertje
[4] https://github.com/google-research/bert
[5] https://github.com/TurkuNLP/FinBERT
[6] https://deepset.ai/german-bert
[7] https://github.com/dccuchile/beto
[8] https://github.com/google-research/bert
[9] https://github.com/mpsilfve/finer-data
[10] https://github.com/TurkuNLP/turku-ner-corpus

Tokens	English	German	Spanish	Dutch	Finnish
Train	203,621	206,931	264,715	202,644	342,924
Development	51,362	51,444	52,923	37,687	31,872
Test	46,435	51,943	51,533	68,875	67,425

Entities	English	German	Spanish	Dutch	Finnish
Train	23,499	11,851	18,798	13,344	27,026
Development	5,942	4,833	4,352	2,616	2,286
Test	5,648	3,673	3,559	3,941	5,129

Table 1: Key statistics of the NER data sets

All of the NER datasets define separate training, development and test sets, and we follow the given subdivision for each. The training sets for each language are used for fine-tuning the corresponding BERT model for NER, development sets are used for evaluation in hyperparameter selection, and the test sets are only used in final experiments for evaluating models trained with the selected hyperparameters. As previous studies vary in whether to combine development data to training data for training a final model, we report also results where models are trained with a combined training and development set for final test experiments. The datasets for the CoNLL shared task languages contain four different classes of named entities: Person (PER), Organization (ORG), Location (LOC) and Miscellaneous (MISC). The Finnish NER datasets also use the PER, ORG, and LOC types along with three others, Product (PROD), Event (EVENT), and Date (DATE). For implementation purposes we converted all the datasets to the same format prior to experiments: The character encoding of each file was converted to UTF-8, and the NER labelling scheme was converted to IOB2 (Ratnaparkhi and Marcus, 1998) also for corpora that were originally in the IOB scheme (Ramshaw and Marcus, 1995). By contrast to the older IOB scheme, in the IOB2 scheme the label for the first token of a named entity is always marked with a B-prefix (e.g. B-PER), even if the previous token is not part of a named entity. The key statistics for the NER datasets are presented in Table 1. Finally, we note that all the datasets except CoNLL'02 Spanish provide information on document boundaries using special `-DOCSTART-` tokens at the start of each new document.

4 Methods

As the starting point for exploring the cross-sentence information for NER using BERT, we use a NER pipeline implementation introduced by Virtanen et al. (2019) that closely follows the straightforward approach presented by Devlin et al. (2018). Here, the last layer of the pre-trained BERT model is followed by a single time-distributed dense layer which is fine-tuned together with the pre-trained BERT model weights to produce the softmax probabilities of NER tags for input tokens. No modelling of tag transition probabilities or any additional processing to validate tag sequences is used.

In our implementation, exactly one example is constructed for each sentence of the corpus unless the sentence is so long that it does not fit to the maximum sequence length[11]. The sentence is placed at the beginning of the BERT window and following sentences from the corpus are used to fill the window (up to the maximum sequence length), with special separator (`[SEP]`) tokens separating the sentences. Partial sentences are used to fill up the BERT examples. As a special case, the sentences used for filling the window for the last sentences in input data are picked by wrapping back to the beginning of the corpus. This approach creates situations where some input samples contain sentences from different original documents, if the documents were next to one another in the corpus. For this reason, we also implemented documentwise wrapping of sentences if the input data had document boundaries marked with `-DOCSTART-` tokens. We used this information to build input samples by filling the sentences at the end of one document with the sentences from beginning of that same document instead of the next sentences in the original data. In this case only full sentences are added to each input sample, and

[11] In this special case the long sentence is split to produce multiple input sequences that are considered as sentences for the rest of the implementation.

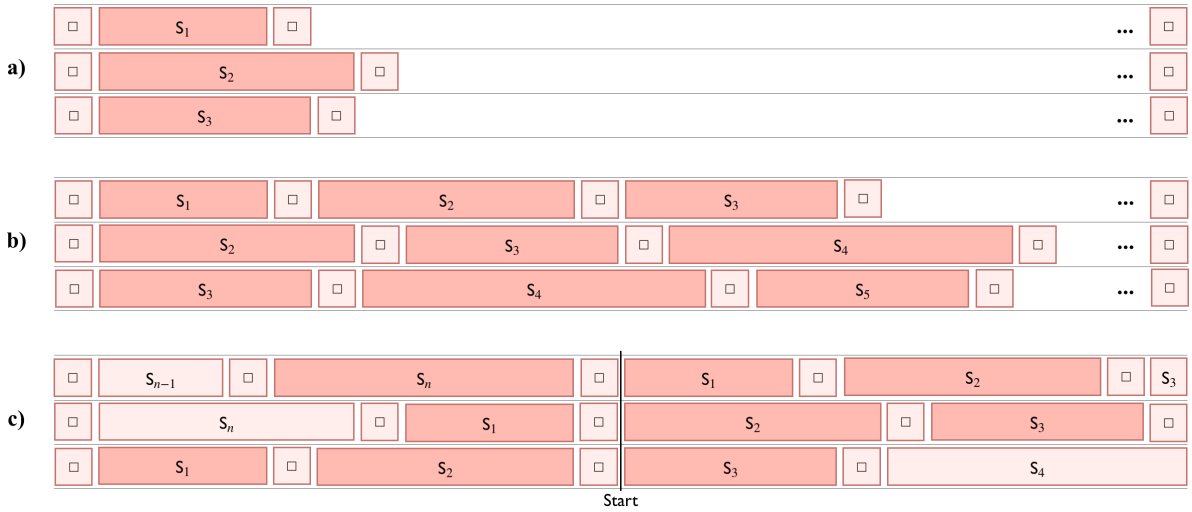


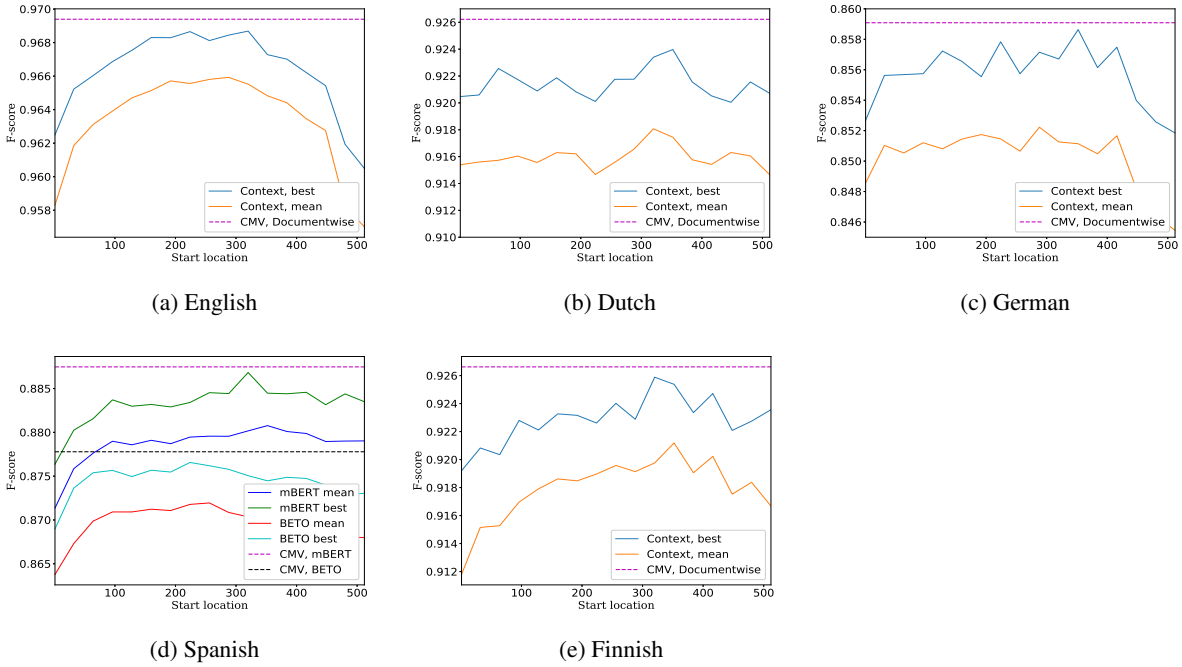
Figure 1: Illustration of various input representations for sequence labelling tasks. a) One sentence per example (*Single*), b) including following sentences (*First, CMV*), c) including preceding and following sentences (*Sentence in context*). CMV combines predictions for the same sentence (e.g. S_2 in b) in various positions and contexts. The empty square (□) stands for special separator symbols (e.g. `[CLS]`, `[SEP]` and `[PAD]` for BERT); a light background color is used to represent special symbols and incomplete sentences in c).

padding (`[PAD]`) tokens are used to fill empty space if the next sentence in the input data does not fit into the window as demonstrated in (Figure 1b).

Constructing inputs in this way implies that the same sentences from the original data occur in different positions and with varying (sizes of) left and right contexts in different samples. We wanted to examine the predictions in different contexts more closely to see if there are consistent effects on tag prediction quality depending on the starting position of a sentence inside a context. One challenge here was how to consistently measure performance with different contexts: sentences are of different lengths, and as they are added to input samples, the beginning of the window was only place where the starting locations of sentences would align. Also, the number of sentences that fit into the window vary substantially. For this reason, it is not possible e.g. to always pick the N*th* sentence to study as there are no guarantees one will exist in all examples. To address this issue and build input samples for testing predictions at different locations, we placed the sentence of interest to start at a specified location inside the window, and filled the window in both directions with sentences before / after the sentence of interest in the original data. We tested the starting positions of the sentence of interest from 1 (0 being the `[CLS]` token) up to the maximum sequence length (512 wordpieces) with intervals of 32 wordpieces. If the sentence of interest was longer than the space between a starting position and the maximum sequence length, the starting position for that particular sentence was moved backwards to fit the sentence in the window.

Ensembles of classifiers are commonly used to improve classification performance at various tasks, and it seems reasonable to assume that predictions for the same input sentences in different positions and contexts create an ensemble-like construct. This is not an ensemble in the conventional sense, as the number of predictions we get for each sentence varies. We evaluate two different variations combining the results from multiple predictions in different contexts. The first approach is to assign labels to sentences in each location first, and then take a majority vote of the assigned labels. The second approach is to add together the softmax probabilities of predictions in different contexts, and then take the argmax of the sum. For simplicity, we here term both Contextual Majority Voting (CMV) as they are variations of the same underlying idea. The implementation uses only predictions of tokens in whole sentences, not ones in partial sentences that may appear in input examples.

For fine-tuning the pre-trained BERT models, we largely follow the process introduced in (Devlin et al., 2018). We use the maximum sequence length of 512 in all experiments to include maximal cross-



(a) English (b) Dutch (c) German

(d) Spanish (e) Finnish

Figure 2: NER performance on development set measured with CMV and in different sentence starting locations. The lower curves show mean performance over whole hyperparameter range, and the upper curves the results with the best hyperparameters (mean of 5 repetitions) for each location. The flat dashed lines show the best CMV results.

sentence context, the Adam optimizer (Kingma and Ba, 2014) ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-6$) with warmup of 10% of samples, linear learning rate decay, a weight decay rate of 0.01, and norm clipping on 1.0. Sample weights are used for inputs so that the special tokens `[CLS]` and `[PAD]` are given zero weight and everything else 1 when calculating the loss (sparse categorical cross-entropy).

We select hyperparameters with an exhaustive search of the grid proposed by Devlin et al., modified to skip batch size 32 and add batch sizes 2 and 4 instead as our initial experiments indicated better performance with smaller batch sizes. That is, the grid search is done over the following parameter ranges:

- Learning rate: 2e-5, 3e-5, 5e-5
- Batch size: 2, 4, 8, 16
- Epochs: 1, 2, 3, 4

We repeated each experiment 5 times with every hyperparameter combination. The best hyperparameters were selected based on the mean of exact mention-level F1 scores, as evaluated against the development set using a Python implementation of the standard conlleval evaluation script.

As a reference we use a BERT model which is fine-tuned using only single sentences from the input data. For this baseline, predictions are also made on the basis of single sentences (see Figure 1a).

5 Results

Based on initial development set results, we decided to focus only on CMV using examples constructed document-wise of the variations of this method (see Section 4). The exception here is the Spanish CoNLL dataset, for which document boundary information was not available. Further, as the differences between CMV variations were found not to be large, we decided to only consider the variant that first assigns labels and then votes between the labels.

The effect of the sentence of interest starting location and the effect of CMV method on development data is illustrated in Figure 2. Our initial expectation was that placing the sentence of interest near

	Precision	Recall	F1	F1 train+dev
English, CMV	93.06 (0.25)	**93.78** (0.08)	93.42 (0.12)	93.57 (0.33)
English, First	**93.15** (0.15)	93.73 (0.04)	**93.44** (0.06)	**93.74** (0.25)
English, Single	91.12 (0.25)	92.28 (0.23)	91.70 (0.24)	91.94 (0.15)
Dutch, CMV	**93.12** (0.26)	93.26 (0.18)	93.19 (0.21)	**93.49** (0.23)
Dutch, First	93.03 (0.65)	**93.38** (0.38)	**93.21** (0.51)	93.39 (0.26)
Dutch, Single	91.57 (0.35)	91.49 (0.41)	91.53 (0.37)	91.92 (0.30)
Finnish, CMV	**92.91** (0.18)	**94.42** (0.13)	**93.66** (0.13)	**93.78** (0.26)
Finnish, First	92.56 (0.14)	94.24 (0.08)	93.39 (0.10)	93.65 (0.26)
Finnish, Single	90.74 (0.10)	92.11 (0.24)	91.42 (0.16)	91.97 (0.21)
German, CMV	**86.91** (0.31)	**84.38** (0.32)	**85.63** (0.30)	**87.31** (0.27)
German, First	86.37 (0.39)	84.07 (0.10)	85.21 (0.22)	86.91 (0.11)
German, Single	85.55 (0.20)	81.81 (0.31)	83.64 (0.21)	85.67 (0.25)
Spanish, CMV	**87.80** (0.25)	**87.98** (0.18)	**87.89** (0.21)	**87.97** (0.21)
Spanish, First	86.71 (0.31)	87.41 (0.28)	87.06 (0.28)	87.27 (0.25)
Spanish, Single	87.43 (0.53)	87.90 (0.34)	87.66 (0.43)	87.52 (0.41)
S-mBERT, CMV	**87.25** (0.50)	**88.67** (0.46)	**87.95** (0.47)	**88.32** (0.26)
S-mBERT, First	86.92 (0.40)	87.88 (0.44)	87.40 (0.42)	87.54 (0.25)
S-mBERT, Single	87.19 (0.28)	87.81 (0.26)	87.50 (0.26)	87.57 (0.29)

Table 2: NER results for different methods and languages (standard deviation in parentheses).

the middle of the sequence would generally yield the best performance. However, while this effect can be observed e.g. for English (Figure 2a), the pattern does not hold in all cases, although in most cases performance does improve when moving the starting position away from either end of the context window. The problem was that the performance in the middle of the context did not appear to be stable enough to pick a reliable starting position to look at prediction time. This can be seen in the figure 2 where the results for different starting locations tend to vary without a clear central optimum. The results for Dutch (Figure 2b) deviated the most from our expectations, and a possible reason for this was later found from the source data: the sentence order of the documents inside the original Dutch language data set has been randomized for copyright reasons. To test if randomizing the sentence order of documents has an effect on results, we tested this with other languages. However, in our initial experiments randomizing sentences inside each document did not result in significant performance drop on any of the tested languages.

The final test set results for models trained with the best hyperparameter combinations found using the development sets are summarized in Table 2. We report precision, recall and F1-score for models trained only on the training dataset, and additionally F1-scores for models trained with combined training and development sets using the same hyperparameters. For each language/BERT model pair, we report performance for the baseline using only a single sentence per window (Single), the approach where sentences from the following context are included but only predictions for the first sentence in each window are used (First), and, finally, performance with CMV (see also Figure 1).

These results show that BERT NER predictions systematically benefit from access to cross-sentence context. For all tested languages except Spanish, models that are fine-tuned and tested with samples containing context outperform models which do not use any context and are relying only on single sentences. What is not directly seen from Table 2 is that generally the results with the method First outperform the results with the method Single, and similarly the method CMV generally outperforms the method First. Both English and Dutch seem to perform well with the method First and for Spanish the method Single also performs well. One thing to note is that English and Dutch results with CMV outperform the method First with the hyperparameters that produced the best result for the method First. However, the final results for CMV just were not as good with the hyperparameters that produced the best performance for CMV on the development data.

Model	Our F1	Our F1 (t+d)	Current BERT	Current SOTA
English	93.44	**93.74**	93.47 (Liu et al., 2019b)	93.5 (Baevski et al., 2019)
Dutch	93.21	**93.49**	90.94 (Wu and Dredze, 2019)	92.69 (Straková et al., 2019)
Finnish	93.66	**93.78**	93.11 (Luoma et al., 2020)	93.11 (Luoma et al., 2020)
German	85.63	87.31	82.82 (Wu and Dredze, 2019)	**88.32** (Akbik et al., 2018)
Spanish	87.89	87.97	88.43 (Cañete et al., 2020)	**89.72** (Conneau et al., 2020)
Spanish, mBERT	87.95	88.32	88.43 (Cañete et al., 2020)	**89.72** (Conneau et al., 2020)

Table 3: NER result comparison to the state of the art.

To further evaluate the performance of CMV method, we checked the results of each fine-tuned model on the development set during hyperparameter search. There were 48 hyperparameter combinations to evaluate for each model. For English, German, Spanish and Finnish, the CMV method outperformed the method First for every hyperparameter combination when calculating the results as the mean of mention-level F1 scores from 5 repetitions. For Spanish this includes both the experiments with the Spanish monolingual model as well as the experiments with the multilingual model. The only exception to this were the results on Dutch, for which CMV outperformed the method First in 41 cases out of 48. The fact that sentences in Dutch data are in randomized order may contribute to this. In total, the CMV method improved the results over method First in 281 cases out of 288. In the same fashion, we evaluated the difference in performance between the method Single and the method First evaluated against the development set. The method First outperformed the method Single for every hyperparameter combination for every tested language.

In Table 3 we compare the results using cross-sentence context with current the state-of-the-art in NER for the languages studied here. We are able to establish a new state-of-the-art result for three languages, English, Dutch and Finnish, as well as improve the best BERT-based score on German. These results benefit from using the combined training and development set in final model training. The previous state-of-the-art is also surpassed on Dutch and Finnish when only the training set is used for the final model. On Spanish our results fall slightly below the reported state-of-the-art. Perhaps somewhat surprising was that multilingual BERT outperformed the dedicated Spanish language BERT model, failing to replicate the results of Cañete et al. (2020), who reported that the Spanish model outperformed that of Wu and Dredze (2019), who had previously reached the best Spanish BERT performance using multilingual BERT. Despite this minor discrepancy, we find that both the simple approach of including following sentences as context as well as CMV are very effective, allowing a straightforward BERT NER model to achieve state-of-the-art performance with only a few modifications of the representation.

6 Discussion

The results presented here are, as far as we know, the first systematic study on how cross-sentence information can be utilized with BERT for NER, and the methods presented here form a good starting point for discussion and further research into the subject. Contextual Majority Voting is straightforward to implement in existing BERT-based systems as the actual model and associated infrastructure is not modified. It is quite probable that similar ways of including cross-sentence information or majority voting structures may be beneficial with other attention-based models as well. The computational overhead for the required pre- and postprocessing of the samples is very modest, but increasing the maximum sequence length in fine-tuning e.g. from 128 to 512 to fit more sentences in one sample does come with a tradeoff of increased computational cost.

One aspect deserving more study is how prediction performance is affected if sentences are not repeated, or repeated fewer times, in examples during prediction. Reducing or entirely avoiding repetition would allow for more efficient use of the model while still providing context for sentences, which might be a reasonable compromise between performance and computational efficiency for large-scale practical applications. A further possibility for future research would be to explore weighted majority voting. Our results lend some support to the idea that predictions made for tokens around in the center of the window are generally more reliable than predictions for tokens near its edges, where context is limited on one side of the token. Providing higher weight to predictions in the middle of the sequence could potentially

help further improve the performance of the aggregation approach. Another aspect for future work would be to study the effect of the context and sentence order. Our preliminary tests with randomized sentence order from same documents showed minimal effect on performance. Is it enough to have context from the same document? Would the situation change drastically if random sentences from the whole input data were used instead? Finally, the incorporation of transition probabilities or other processing to check tag sequences for illegal transitions would likely improve performance further.

7 Conclusions

We have presented a comprehensive evaluation of the effect of including cross-sentence context for named entity recognition with BERT and introduced a simple and easy-to-implement approach for the task using majority voting. The proposed method established new state-of-the-art results in named entity recognition for three languages and is near the state-of-the-art for two other languages, demonstrating how simple ideas may boost the performance of even very strong models.

We release all methods implemented in this work under open licenses from `https://github.com/jouniluoma/bert-ner-cmv`.

Acknowledgements

We wish to thank the CSC – IT Center for Science, Finland, for generous computational resources. This work was funded in part by the Academy of Finland.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. 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 724–728, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven pretraining of self-attention networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Pratyay Banerjee, Kuntal Kumar Pal, Murthy Devarakonda, and Chitta Baral. 2019. Knowledge guided named entity recognition for biomedical text.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *to appear in PML4DC at ICLR 2020*.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357370, Dec.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12(null):24932537, November.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard 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, Online, July. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 168–171.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Vijay Krishnan and Christopher D. Manning. 2006. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1121–1128, Sydney, Australia, July. Association for Computational Linguistics.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. Flaubert: Unsupervised language model pre-training for french. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France, May. European Language Resources Association.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020a. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, page 11.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020b. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online, July. Association for Computational Linguistics.

Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019a. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307, Florence, Italy, July. Association for Computational Linguistics.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019b. Gcdt: A global context enhanced deep transition architecture for sequence labeling. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical contextualized representation for named entity recognition. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8441–8448. AAAI Press.

Jouni Luoma, Miika Oinonen, Maria Pyyknen, Veronika Laippala, and Sampo Pyysalo. 2020. A broad-coverage corpus for finnish named entity recognition. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4615–4624, Marseille, France, May. European Language Resources Association.

Andrei Mikheev, Claire Grover, and Marc Moens. 1998. Description of the LTG system used for MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 78–86, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Adwait Ratnaparkhi and Mitchell P. Marcus. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, USA. AAI9840230.

Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. 2019. A Finnish news corpus for named entity recognition. *Language Resources and Evaluation*, pages 1–26.

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, Florence, Italy, July. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 -*.

Erik F. Tjong Kim Sang, Walter Daelemans, Herve Dejean, Rob Koeling, Yuval Krymolowski, Vasin Punyakanok, and Dan Roth. 2000. Applying system combination to base noun phrase identification. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

Erik F. Tjong Kim Sang. 2002. Introduction to the conll-2002 shared task. *proceeding of the 6th conference on Natural language learning - COLING-02*.

Hans Van Halteren, Jakub Zavrel, and Walter Daelemans. 2001. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational Linguistics*, 27(2):199–229.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. *arXiv:1912.09582 [cs]*, December.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.