# Improving Stability of Transformer-based Named Entity Recognition Models with Combined Data Representation

Michał Marcińczuk *

Posted Date: 28 September 2023

doi: 10.20944/preprints202309.1859.v1

Keywords: named entity recognition; deep learning; transformers; data augmentation; pre-trained language models; PLM

*Article*

# Improving Stability of Transformer-Based Named Entity Recognition Models with Combined Data Representation

**Michał Marcińczuk** [1,2]

[1] CodeNLP; marcinczuk@gmail.com
[2] Wrocław University of Science and Technology; michal.marcinczuk@pwr.edu.pl
* Correspondence: marcinczuk@gmail.com

**Abstract:** This study leverages transformer-based models and focuses on data representation strategies in the named entity recognition task, including "single" (one sentence per vector), "merged" (multiple sentences per vector), and "context" (sentences joined with attention to context). Performance analysis reveals that models trained with a single strategy may not perform well on different data representations. A combined training procedure is proposed to address this limitation, using all three strategies to enhance the stability and adaptability of the model. The results of this approach are presented and discussed for various datasets for four languages (English, Polish, Czech, and German), demonstrating the effectiveness of the combined strategy.

**Keywords:** named entity recognition, deep learning, transformers, data augmentation

## 1. Introduction

Named Entity Recognition (NER) is a fundamental task in natural language processing, focusing on identifying word sequences that belong to specific entity categories, such as people, locations, organizations, objects, and events. However, what constitutes a named entity varies across different applications and languages. This study delves into the challenges posed by NER, particularly due to the presence of proper names that reference entities without inherently providing descriptive information.

The recognition of named entities poses several challenges. One challenge arises because many named entities are proper names, which are rigid designators [1]. Proper names serve as references to entities and do not inherently provide descriptive information about those entities. Consequently, identifying named entities requires prior knowledge that a particular term qualifies as a named entity. Furthermore, the set of possible named entities is vast and limitless.

Several approaches can be used to train a model to recognize a term as a named entity. These include training on annotated datasets that identify named entities, incorporating a common sense knowledge base about the world, or inferring characteristics from the training data. For example, consider the sentence "Mark works in Xax." The term "Xax" is probably a named entity because it is capitalized. However, it may require additional information to determine its semantic category, such as whether it refers to a location (city or country) or a company name. Subsequent sentences like "He loves this city." could provide contextual clues, clarifying that "Xax" is indeed the name of a city. It is worth noting that the less information available from the input text, the greater the reliance on information contained in the training dataset or external sources.

Current state-of-the-art methods in the field of named entity recognition are based on pre-trained language models in a transformers architecture [2–5]. Transformer-based models outperform approaches like LSTM neural networks or conditional random fields (CRF).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 September 2023 doi:10.20944/preprints202309.1859.v1



## 2. Datasets

We used in our research five datasets for four languages: English, Polish, German, and Czech. The summary of the datasets is presented in Table 1. The following subsections contain a brief description of each dataset.

**Table 1.** List of dataset used in the research

| Corpus | Language | Sentences | Tokens | Annotations | Categories | Nested |
|---|---|---|---|---|---|---|
| CoNLL 2003 | English | 20 744 | 301 418 | 35 089 | 4 | No |
| GermEval 2014 | German | 31 300 | 591 005 | 40 941 | 12 | Yes |
| CNEC 2.0 | Czech | 8 992 | 199 965 | 28 727 | 47 | Yes |
| NKJP | Polish | 153 982 | 2 180 611 | 154 292 | 14 | Yes |
| KPWr (n82) | Polish | 18 282 | 303 678 | 17 716 | 82 | Yes |

### *2.1. CoNLL 2003 NER (English)*

CoNLL 2003 NER[1] [6] is a benchmark dataset for Named Entity Recognition in English. It consists of news articles from various sources, annotated with labeled entities such as person names, locations, organizations, and miscellaneous entities. The dataset contains four coarse-grained categories of entities: person, location, organization, and miscellaneous.

### *2.2. GermEval 2014 (German)*

GermEval 2014[2] [7] is a benchmark dataset for Named Entity Recognition for German. The dataset consists of samples from German Wikipedia and News Corpora. It contains four main NER categories with sub-structure and annotating embeddings among named entities.

### *2.3. CNEC 2.0 (Czech)*

CNEC 2.0[3] [8] is a benchmark dataset for Named Entity Recognition for Czech. The named entities are classified according to a two-level hierarchy, which defines 46 fine-grained categories.

### *2.4. NKJP (Polish)*

NKJP (The National Corpus of Polish)[4] [9] is a collection of texts of various genres, including classic literature, daily newspapers, specialist periodicals and journals, conversation transcripts, and various short-lived and Internet texts. The dataset contains 11 categories of named entities. Annotations can be nested multiple times.

The dataset has no official split into train, dev, and test subsets. Thus, we used our custom division. We used 80% of the data as a training subset and the remaining as a test subset.

### *2.5. KPWr (Polish)*

KPWr[5] [10] is a collection of Polish texts distributed under an open license. It contains more than 1,400 short excerpts from texts of various genres. The annotation schema defines 82 fine-grained categories of named entities.

---

1 https://www.clips.uantwerpen.be/conll2003/ner/
2 https://sites.google.com/site/germeval2014ner/data
3 https://ufal.mff.cuni.cz/cnec/cnec2.0
4 https://nkjp.pl/?lang=1
5 https://clarin-pl.eu/dspace/handle/11321/722

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 September 2023 doi:10.20944/preprints202309.1859.v1



## 3. Research problem statement

### *3.1. Named entity recognition with transformers*

To model the problem of named entity recognition with transformers, we follow the original approach presented by [11]. This approach represents the recognition of named entities as a sequence classification task. The input to the model consists of a sequence of subtokens produced by the tokenizer. The model produces a sequence of labels representing the boundaries of named entities. The labels are aligned with words, meaning only the first subtoken of each word is used as the input to the token-level classifier over the label set. The named entity labels are encoded using the IOB2 schema (Inside, Outside, and Begin). We followed the procedure presented by [12] to encode nested entities. Every nested entity is decoded separately using the IOB2 schema in this approach. Then, all the entity-level labels are concatenated for each token into a single word-level label. For instance, a word that begins a person's name and is inside an organization name will be encoded as `I-ORG#B-PER`, where # jest a label separator.

The neural network architecture consists of two main elements: pre-trained language model (PLM) embeddings and a classifier (see Figure 1). The PLM part generates a context-aware representation of each word (a vector of the first word's subtoken). The classifier part outputs one of the IOB2 labels. The label set depends on a given dataset's categories of named entities. In the post-processing, the sequences of IOB2 labels are wrapped into continuous annotations.

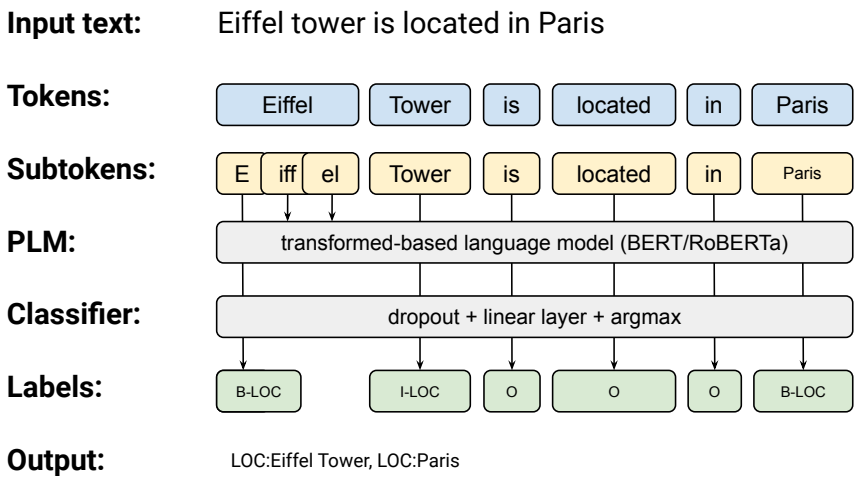


**Figure 1.** The neural network architecture for named entity recognition.

The model processes the text in small batches, each containing a fixed number of subwords. The number of subwords depends on the architecture of PLM. The first PLMs, such as BERT and RoBERTa, used a size of 512 subwords. There are also PLM that offer much longer sequences. For instance, BigBird [13], and Longformer [14] can handle sequences up to 4096 subtokens. The transformer architecture uses a multi-head attention mechanism, which allows it to learn long-range dependencies between subtokens in a sequence. Therefore, the strategy for dividing the text into fragments is crucial for data processing. The goal is to divide the text into fragments that are small enough for the transformer to learn the long-range dependencies but large enough to capture the important information in the text. In the next section, we discuss different strategies of data representation.

### *3.2. Data representation*

The transform model's data representation considers two factors: internal context (how the document is split into smaller pieces) [11] and external context (out-of-document knowledge

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 September 2023 doi:10.20944/preprints202309.1859.v1



concatenated to the input vector) [15]. Our research focuses only on document-based representation, which is based solely on the input document and does not use any external knowledge. Strategies are visualized in Figure 2.

1. **Single** — each vector contains exactly one sentence. If a sentence is longer than the maximum number of subtokens, it is split into several non-overlapping vectors.
2. **Merged** — each vector can contain more than one sentence. The vectors are filled with sentences until the limit of maximum subtokens is reached. This approach provides high-speed improvements, as it reduces the number of vectors to process.
3. **Context** — sentences within a document are joined into a single sequence. The sentences are separated with a special token representing the sentence boundary. Then, the sequence is divided into windows of size smaller than the maximum sequence size. A vector is generated for every window by joining the windows with some subtokens before and after the windows. The added subtokens serve as a context and are masked out. This means that they are used only to calculate the attention but are not used to train the linear token classifier.

   The advantage of this representation is the efficient context usage for embedding generation with the attention mechanism. Each subtoken has at least N token before and after. The N is the minimal size of the subtokens added before and after the window. This strategy also gives the best performance already proved in other research [11].

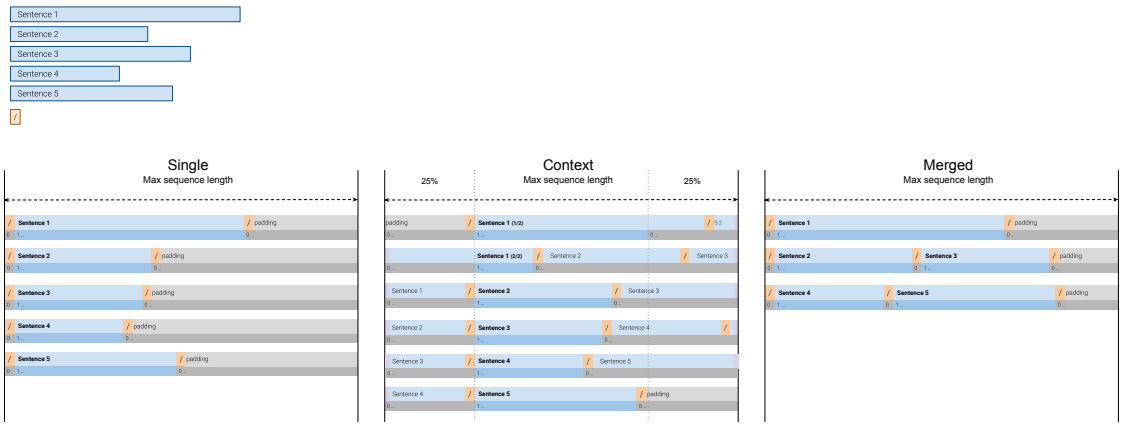


**Figure 2.** An overview of the sequence generation strategies

### *3.3. Training parameters*

For each experiment, we used the same set of parameters. The only difference was the pre-trained language model (PLM), which depends on the language. The parameters are summarized in Table 2.

**Table 2.** Training parameters of the neural network

| Parameter | Value |
|---|---|
| Sequence length | 256 |
| Dropout | 0.2 |
| Epochs | 20 |
| Learning rate | 5e-6 |
| PLM | - *xlm-roberta-large* for English, Czech, and German<br>- *allegro/herbert-large-cased* for Polish |
| Optimizer | AdamW |
| Scheduler | Liner without warmup |

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 September 2023 doi:10.20944/preprints202309.1859.v1



### 3.4. Performance discrepancy between strategies

Table 3 presents the evaluation of the models trained using different strategies of vector representation, i.e., *single*, *merged*, and *context* (see Section 3.2). We can observe that the best performance for each corpus was obtained for the *context* representation. The difference between the second-best score varies from 0.3 pp for GermEval 2014 to 16.48 pp for KPWr (n82).

**Table 3.** An average of F1 from 5 runs on the tune parts of the datasets using the same setup for training and testing.

| Corpus | Single | | Merged | | Context | |
|---|---|---|---|---|---|---|
| | F1 [%] | $\sigma$ | F1 [%] | $\sigma$ | F1 [%] | $\sigma$ |
| CoNLL 2003 | 96.27 | 0.15 | 96.01 | 0.31 | **96.79** | 0.26 |
| GermEval 2014 | 87.18 | 0.18 | 85.91 | 0.62 | **87.48** | 0.40 |
| CNEC 2.0 | 83.85 | 0.18 | 80.85 | 0.51 | **84.88** | 0.09 |
| NKJP | 92.21 | 0.14 | 91.41 | 0.17 | **92.32** | 0.11 |
| KPWr (n82) | 40.71 | 34.78 | 61.45 | 1.60 | **77.93** | 0.26 |

Looking at Table 3, it is easy to conclude that the best models are the ones with the *context* representation. However, we will obtain significantly different results when we take the model trained with the context representation and evaluate it using the *single* representation. The change in inference simulates a different model utilization scenario — the model is used to process short text snippets (single sentences) rather than long text fragments. We can observe a loss in performance from 3.50 pp for NKJP to 29.37 for CoNLL 2003. We can also observe a much larger variance in the results. The standard deviation varies from 1.20 to 40.28.

One can say that the problem is the lack of context – the *context* representation provides more information than a single sentence. However, when we compare the results for *Context/Single* (from Table 4) with *Single* (from Table 3), we can observe a similar loss in performance (see Table 5). For four corpora, we obtained significantly lower results. Only for KPWr (n82), we obtained an improvement of 9.73. At the same time, the variance for the Single/Single model is 34.78, which indicates that the model was unstable. The lack of stability could be due to the small size of the training dataset compared to the number of categories of named entities and the number of distinct named entities in the testing dataset (see Table 1).

**Table 4.** An average of F1 from 5 runs on the tune parts of the datasets using different setups on training and inference – context for training and single for inference.

| Training/Inference | Context/Context | | Context/Single | | Diff |
|---|---|---|---|---|---|
| Corpus | F1 [%] | $\sigma$ | F1 [%] | $\sigma$ | F1 [%] |
| CoNLL 2003 | **96.79** | 0.26 | 67.42 | 40.28 | -29.37 |
| GermEval 2014 | **87.48** | 0.40 | 72.92 | 6.91 | -14.56 |
| CNEC 2.0 | **84.88** | 0.09 | 69.49 | 4.16 | -15.39 |
| NKJP | **92.32** | 0.11 | 88.82 | 1.20 | -3.50 |
| KPWr (n82) | **77.93** | 0.26 | 50.44 | 8.82 | -27.39 |

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 September 2023 doi:10.20944/preprints202309.1859.v1



**Table 5.** An average of F1 from 5 runs on the tune parts of the datasets using different setups on training and inference – context for training and single for inference.

| **Training/Inference** | **Single/Single** | | **Context/Single** | | **Diff** |
|---|---|---|---|---|---|
| **Corpus** | F1 [%] | $\sigma$ | F1 [%] | $\sigma$ | F1 [%] |
| CoNLL 2003 | **96.27** | 0.15 | 67.42 | 40.28 | -28.85 |
| GermEval 2014 | **87.18** | 0.18 | 72.92 | 6.91 | -14.26 |
| CNEC 2.0 | **83.85** | 0.18 | 69.49 | 4.16 | -14.36 |
| NKJP | **92.21** | 0.14 | 88.82 | 1.20 | -3.39 |
| KPWr (n82) | 40.71 | 34.78 | **50.44** | 8.82 | +9.73 |

### *3.5. Observations*

Based on our initial research, we formulated the following two observations:

1. Model trained solely on a dense (context) or sparse (sentence/merged) representation may not perform well on the opposite. The discrepancy depends on the dataset.
2. No single strategy is stable across all tested corpora and dense/sparse representation.

We argue that the two limitations can be neglected by combining all strategies during the training process. This will lead to a more stable model that can yield better performance despite the input strategy on the inference. The results of our research are presented in the next section.

## 4. Combined data representation

We propose a training procedure that uses all three data representation strategies presented in Section 3.2 (called *union*). The rationale behind this idea is that a model trained on a given representation is biased toward that representation. This was shown in Section 3.4 that a model trained on data with context performs poorly on data without context. At the same time, a model trained on data without context achieved good performance on data without context. By combining all strategies, we perform a form of data augmentation. Each sample is used three times but with slightly different contexts. Due to the multi-head attention mechanism, this impacts the vector representation of each word. In this way, the model becomes more resilient to context modifications.

In Appendix A (Tables A1–A5), we present detailed results for each combination of data representation during training (rows) and inference (columns). For training, we include each strategy (*single*, *merged*, and *context*) and the combined one (*union*). For inference, we included *single*, *merged*, and *context*. We omit *union* on inference, as it is not suitable for the inference. The last column of each row contains the average performance for all three strategies. For each dataset, except for NKJP, we reported results for the dev and test subsets. In the analysis, we will focus only on the test subset, as it was available for each dataset.

By using the *union* strategy on training, we obtained at least the same performance for each dataset as for any other strategy. The highest improvement of 2.65pp was obtained for the KPWr dataset — from 76.80% vs. 79.45%. The smallest improvement we obtained for the CoNLL 2003 was only 0.15pp. Table 6 presents the differences for all datasets.

**Table 6.** Comparision of the training strategies — best F-measure for any *single*, *merged*, *context* vs. F-measure for *union* on the test subsets.

| **Dataset** | **Best of *single*, *merged*, *context*** | ***Union*** | **Diff** |
|---|---|---|---|
| CoNLL 2003 | 93.54 | 93.69 | +0.15 |
| GermEval 2014 | 87.57 | 88.04 | +0.57 |
| CNEC 2.0 | 82.70 | 83.82 | +1.12 |
| NKJP | 92.32 | 92.78 | +0.46 |
| KPWr (n82) | 76.80 | 79.45 | +2.65 |

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 September 2023 doi:10.20944/preprints202309.1859.v1



We can observe a much higher improvement when we compare the average performance of different inference strategies. As Section 3.5 points out, a model trained with one strategy may perform poorly using another strategy on inference. This time, the improvement for CoNLL 2003 was 1.27 pp and for KPWr — 11.17 pp. The summary of the differences is presented in Table 7.

**Table 7.** Comparision of the training strategies — best average F-measure for any *single*, *merged*, *context* vs. average F-measure for *union* on the test subsets.

| Dataset | Avg. best of *single*, *merged*, *context* | Avg. *union* | Diff |
|---|---|---|---|
| CoNLL 2003 | 91.68 | 92.98 | +1.27 |
| GermEval 2014 | 85.54 | 87.77 | +2.23 |
| CNEC 2.0 | 79.51 | 83.69 | +4.18 |
| NKJP | 92.04 | 92.77 | +0.73 |
| KPWr (n82) | 67.63 | 78.80 | +11.17 |

## 5. Comparision with SOTA

This section compares the obtained performance for the *union* strategy with the SOTA models. Table 8 summarizes the results for all datasets. For all datasets, except CoNLL 2003, we outperformed the SOTA results.

For CoNLL 2003, we obtained the average F-measure of 93.69, 0.91 pp below the SOTA. However, our results cannot compete with [16] as they combined several pre-trained models to get word representation. In our research, we used only one pre-trained language model.

**Table 8.** Comparision of the *union* strategy with the SOTA models. *union* is an average from five runs.

| Dataset | Metric | *union* | SOTA | Reference |
|---|---|---|---|---|
| CoNLL 2003 | F1 | 93.69 | **94.60** | [16] |
| GermEval 2014 | F1 | **88.04** | 87.69 | [12] |
| CNEC 2.0 | F1 | **83.82** | 83.44 | Web page[6] |
| NKJP | Score | **92.60*** | 88.06 | [17] |
| KPWr (n82) | F1 | **79.45** | 73.28 | [18] |

We have a straightforward comparison for GermEval 2014 and CNEC 2.0, as each model is also based on a single pre-trained model — the same that was used in our research, which is a large variant of XLM-Roberta[7]. For both datasets, we obtained slight improvements of 0.35 and 0.38 for GermEval 2014 and CNEC 2.0, respectively.

To compare our model with SOTA results for the NKJP dataset, we used the PolEval 2018 NER dataset[8]. The corpus was created as a benchmark to evaluate models trained on the NKJP dataset. The score of 92.60 differs from the one reported in Table 6 (92.32), as the evaluation metric was a weighted means of overlap and exact matches. We obtained a significant improvement of 4.54 pp compared to the SOTA model.

## 6. Conclusions

Different strategies exist to represent the training data; none is an obsolete solution. If a model is trained using a dense or sparse vector representation, it may perform poorly on the opposite representation. Our research showed we can benefit from training a model using all available representations. A combination of different strategies lengthens the learning process, but what is more

[7] https://huggingface.co/xlm-roberta-large
[8] http://2018.poleval.pl/index.php/tasks

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 September 2023 doi:10.20944/preprints202309.1859.v1



important, it improves model stability and accuracy without lengthening the inference time. We have conducted a series of experiments on five datasets for four languages (English, Polish, German, and Czech). For each dataset, the proposed *union* strategy performed better than any single strategy (*single*, *merged*, *context*). For 4 out of 5 datasets, we also obtained a performance better then SOTA.

## Appendix A Detailed results for each dataset

**Table A1.** The results of the evaluation on the CoNLL 2004 dataset. The model was trained on the train subset. The reported values are the average F1 scores obtained from five runs with different seeds

| | | Inference | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **single** | | **merged** | | **context** | | **AVG** |
| Subset | Training | F1 | $\sigma$ | F1 | $\sigma$ | F1 | $\sigma$ | |
| Dev | **single** | **96.27** | 0.15 | 94.59 | 0.23 | 95.33 | 0.31 | 95.39 |
| | **merged** | 68.72 | 4.02 | **96.01** | 0.31 | 95.87 | 0.12 | 86.87 |
| | **context** | 67.42 | 40.28 | 95.36 | 0.49 | **96.79** | 0.26 | 86.52 |
| | **union** | **96.39** | 0.12 | **96.64** | 0.16 | **96.95** | 0.16 | **96.66** |
| Test | **single** | **92.43** | 0.17 | 90.82 | 0.62 | 91.79 | 0.59 | 91.68 |
| | **merged** | 60.50 | 3.76 | **93.03** | 0.35 | 92.76 | 0.43 | 82.10 |
| | **context** | 63.44 | 38.44 | 92.02 | 0.35 | **93.54** | 0.29 | 83.00 |
| | **union** | **92.18** | 0.29 | **93.06** | 0.22 | **93.69** | 0.14 | **92.98** |

**Table A2.** The results of the evaluation on the GermEval 2014 dataset. The model was trained on the train subset. The reported values are the average F1 scores obtained from five runs with different seeds

| | | Inference | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **single** | | **merged** | | **context** | | **AVG** |
| Subset | Training | F1 | $\sigma$ | F1 | $\sigma$ | F1 | $\sigma$ | |
| Dev | **single** | **87.18** | 0.18 | 84.59 | 0.90 | 85.76 | 0.79 | 85.84 |
| | **merged** | 57.85 | 6.56 | **85.91** | 0.62 | 85.84 | 0.77 | 76.53 |
| | **context** | 72.92 | 9.91 | **86.32** | 0.51 | 85.84 | 0.40 | 82.24 |
| | **union** | **87.97** | 0.18 | **88.06** | 0.08 | **88.13** | 0.50 | **88.05** |
| Test | **single** | **86.95** | 0.30 | 84.09 | 0.56 | 85.57 | 0.35 | 85.54 |
| | **merged** | 57.65 | 5.75 | **85.67** | 0.27 | 85.96 | 0.46 | 76.43 |
| | **context** | 72.99 | 5.93 | 85.82 | 0.30 | **87.57** | 0.38 | 82.12 |
| | **union** | **87.80** | 0.34 | **87.47** | 0.20 | **88.04** | 0.27 | **87.77** |

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 September 2023 doi:10.20944/preprints202309.1859.v1



**Table A3.** The results of the evaluation on the CNEC 2.0 dataset. The model was trained on the train subset. The reported values are the average F1 scores obtained from five runs with different seeds

| | | Inference | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **single** | | **merged** | | **context** | | **AVG** |
| Subset | Training | F1 | $\sigma$ | F1 | $\sigma$ | F1 | $\sigma$ | |
| Dev | **single** | **83.85** | 0.18 | 78.77 | 4.84 | 81.86 | 1.72 | 81.49 |
| | **merged** | 60.42 | 2.93 | **80.85** | 0.51 | 80.83 | 0.40 | 74.03 |
| | **context** | 69.49 | 4.16 | 83.30 | 0.62 | **84.88** | 0.09 | 79.22 |
| | **union** | **85.53** | 0.18 | **85.67** | 0.15 | **86.31** | 0.15 | **85.84** |
| Test | **single** | **81.98** | 0.30 | 76.73 | 3.89 | 79.83 | 1.64 | 79.51 |
| | **merged** | 60.01 | 2.26 | **79.05** | 0.36 | 79.31 | 0.16 | 72.79 |
| | **context** | 69.53 | 3.24 | 80.44 | 0.61 | **82.70** | 0.19 | 77.56 |
| | **union** | **83.79** | 0.21 | **83.44** | 0.29 | **83.82** | 0.43 | **83.69** |

**Table A4.** The results of the evaluation on the NKJP dataset. The model was trained on the train subset. The reported values are the average F1 scores obtained from five runs with different seeds

| | | Inference | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **single** | | **merged** | | **context** | | **AVG** |
| Subset | Training | F1 | $\sigma$ | F1 | $\sigma$ | F1 | $\sigma$ | |
| Test | **single** | **92.21** | 0.14 | 91.83 | 0.66 | 92.08 | 0.58 | 92.04 |
| | **merged** | 3.40 | 1.72 | 91.41 | 0.17 | **91.52** | 0.11 | 62.11 |
| | **context** | 88.82 | 1.20 | 92.06 | 0.23 | **92.32** | 0.11 | 91.07 |
| | **union** | **92.63** | 0.45 | **92.66** | 0.10 | **92.78** | 0.13 | **92.77** |

**Table A5.** The results of the evaluation on the KPWr dataset. The model was trained on the train subset. The reported values are the average F1 scores obtained from five runs with different seeds

| | | Inference | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **single** | | **merged** | | **context** | | **AVG** |
| Subset | Training | F1 | $\sigma$ | F1 | $\sigma$ | F1 | $\sigma$ | |
| Dev | **single** | **40.72** | 34.78 | 8.31 | 19.79 | 10.48 | 23.14 | 19.84 |
| | **merged** | 6.38 | 1.86 | **61.45** | 1.60 | 61.12 | 1.37 | 42.98 |
| | **context** | 50.44 | 7.82 | 76.85 | 0.32 | **77.67** | 0.26 | 68.32 |
| | **union** | **79.28** | 0.50 | **80.25** | 0.25 | **80.37** | 0.15 | **79.97** |
| Test | **single** | **40.38** | 35.54 | 8.64 | 20.37 | 11.12 | 23.46 | 20.05 |
| | **merged** | 5.70 | 1.23 | **61.00** | 1.53 | 60.36 | 1.70 | 42.36 |
| | **context** | 50.33 | 7.74 | 75.75 | 0.63 | **76.80** | 0.65 | 67.63 |
| | **union** | **77.89** | 0.18 | **79.06** | 0.21 | **79.45** | 0.15 | **78.80** |

## References


1. Kripke, S.A. *Naming and Necessity: Lectures Given to the Princeton University Philosophy Colloquium*; Cambridge, MA: Harvard University Press, 1980.
2. Ye, D.; Lin, Y.; Sun, M. Pack Together: Entity and Relation Extraction with Levitated Marker. *arXiv preprint arXiv:2109.06067* **2021**.
3. Wang, X.; Jiang, Y.; Bach, N.; Wang, T.; Huang, Z.; Huang, F.; Tu, K. Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on

Natural Language Processing (Volume 1: Long Papers); Association for Computational Linguistics: Online, 2021; pp. 1800–1812. https://doi.org/10.18653/v1/2021.acl-long.142.

4. Yamada, I.; Asai, A.; Shindo, H.; Takeda, H.; Matsumoto, Y. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In Proceedings of the EMNLP, 2020.
5. Luoma, J.; Pyysalo, S. Exploring Cross-sentence Contexts for Named Entity Recognition with BERT **2020**. https://doi.org/10.48550/ARXIV.2006.01563.
6. Sang, E.F.T.K.; Meulder, F.D. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, 2003, [arXiv:cs.CL/cs/0306050].
7. Benikova, D.; Biemann, C.; Kisselew, M.; Padó, S. GermEval 2014 Named Entity Recognition Shared Task: Companion Paper. 2014, Workshop proceedings of the 12th edition of the KONVENS conference, pp. 104 – 112.
8. Ševčíková, M.; Žabokrtský, Z.; Straková, J.; Straka, M. Czech Named Entity Corpus 2.0, 2014. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
9. Bańko, M.R.; Górski, R.L.R.; Przepiórkowski, A.R.; Lewandowska-Tomaszczyk, B.R. Narodowy Korpus Języka Polskiego. online.
10. Broda, B.; Marcińczuk, M.; Maziarz, M.; Radziszewski, A.; Wardyński, A. KPWr: Towards a Free Corpus of Polish. In Proceedings of the Proceedings of LREC'12; Calzolari, N.; Choukri, K.; Declerck, T.; Doğan, M.U.; Maegaard, B.; Mariani, J.; Odijk, J.; Piperidis, S., Eds.; ELRA: Istanbul, Turkey, 2012.
11. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019, [arXiv:cs.CL/1810.04805].
12. Marcińczuk, M.; Radom, J. A Single-run Recognition of Nested Named Entities with Transformers. *Procedia Computer Science* **2021**, *192*, 291–297. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021, https://doi.org/https://doi.org/10.1016/j.procs.2021.08.030.
13. Zaheer, M.; Guruganesh, G.; Dubey, A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. Big Bird: Transformers for Longer Sequences, 2021, [arXiv:cs.LG/2007.14062].
14. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer, 2020, [arXiv:cs.CL/2004.05150].
15. Tan, Z.; Huang, S.; Jia, Z.; Cai, J.; Li, Y.; Lu, W.; Zhuang, Y.; Tu, K.; Xie, P.; Huang, F.; et al. DAMO-NLP at SemEval-2023 Task 2: A Unified Retrieval-augmented System for Multilingual Named Entity Recognition. In Proceedings of the Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023); Association for Computational Linguistics: Toronto, Canada, 2023; pp. 2014–2028. https://doi.org/10.18653/v1/2023.semeval-1.277.
16. Wang, X.; Jiang, Y.; Bach, N.; Wang, T.; Huang, Z.; Huang, F.; Tu, K. Automated Concatenation of Embeddings for Structured Prediction. *CoRR* **2020**, *abs/2010.05006*, [2010.05006].
17. Dadas, S.; Protasiewicz, J. A Bidirectional Iterative Algorithm for Nested Named Entity Recognition. *IEEE Access* **2020**, *8*, 135091–135102. https://doi.org/10.1109/ACCESS.2020.3011598.
18. Marcińczuk, M. Fine-Grained Named Entity Recognition for Polish using Deep Learning. In Proceedings of the Proceedings of the PP-RAI'2019 Conference, 2019, pp. 219–222.