# CodeNLP at SemEval-2023 Task 2: Data Augmentation for Named Entity Recognition by Combination of Sequence Generation Strategies

**Michał Marcińczuk** and **Wiktor Walentynowicz**
Wrocław University of Science and Technology
{michal.marcinczuk,wiktor.walentynowicz}@pwr.edu.pl

## Abstract

In the article, we present the CodeNLP submission to the SemEval-2023 Task 2: MultiCoNER II Multilingual Complex Named Entity Recognition. Our approach is based on data augmentation by combining various strategies of sequence generation for training. We show that the extended procedure of fine-tuning a pre-trained language model can bring improvements compared to any single strategy. On the development subsets the improvements were 1.7 pp and 3.1 pp of F-measure, for English and multilingual datasets, respectively. On the test subsets our models achieved 63.51% and 73.22% of Macro F1, respectively.

## 1 Introduction

In this study, we address the topic of named entity recognition from the field of natural language processing. The task is to identify sequences of words in a text that refer to some categories of entities — people, locations, organizations, objects, events, etc. There is no one firm definition of what a named entity is. The definition and the range of categories may vary from one application to another.

Named entity recognition is a challenging task due to several factors. One is that many named entities are proper names, and proper names are rigid designators (Kripke, 1980). Proper names refer to entities. They do not describe the entities by definition. This implies that you should know that a given term is a named entity to identify a named entity. The set of named entities is unlimited. To make a model recognize a term as a named entity, you can train on recognition terms from the annotated dataset, provide some form of a common-sense knowledge base about the world, or infer some characteristics from training data. For instance, in the sample sentence "Mark works in Xax.", the term *Xax* is probably a named entity because it is capitalized. However, semantic categorization might require some additional information about the term, as it might refer to a location (city or country) or a company name. For example, the following sentence, "He loves this city." might clarify that *Xax* is a name of a city. The less information we have from the input text, the more information we need in the training dataset or external sources.

In the paper, we deal with the named entity recognition task called MultiCoNER II (Fetahu et al., 2023b; Malmasi et al., 2022b). The main challenges of this task are: a) fine-grained categories of named entities – 6 main categories and 36 subcategories; b) all texts are lowercase, and c)the dataset consists of short sentences.

## 2 Related Work

Current state-of-the-art methods in the field of named entity recognition are based on either LSTM networks (Yu et al., 2020; Xu et al., 2021), or pre-trained language models in a transformers architecture (Ye et al., 2021; Wang et al., 2021b; Yamada et al., 2020a; Luoma and Pyysalo, 2020).

These methods are characterized by different approaches to improving performance by dealing with specific problems. An essential element is a way in which the processed text is represented. (Wang et al., 2021a) combines different vector representations, while (Ye et al., 2021) propounds an entirely new approach to delimiting areas representing entities. In addition, (Yamada et al., 2020b) proposes an extended presentation of entities in the context. (Li et al., 2020b) describes the use of a special loss function for unbalanced data. Instead, the (Wang et al., 2021b) authors propose to search for additional contexts for under-represented data to improve the quality of the system.

The best-performing methods also vary depending on the dataset on which they are evaluated. Thus, the current best performing solutions (Li et al., 2020b; Ye et al., 2021; Yu et al., 2020; Li et al., 2020a; Xu et al., 2021) on the OntoNotes v5 dataset (Weischedel et al.) are a disjoint set from



*Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1798–1804
July 13-14, 2023 

the best performing solutions (Wang et al., 2021b; Ushio and Camacho-Collados, 2021; Nguyen et al., 2020; Fu et al., 2021; Mayhew et al., 2020) on the WNUT 2017 dataset (Derczynski et al., 2017).

The topics solved in this edition of the SemEval event (Fetahu et al., 2023b) are a continuation of the topics from the previous edition (Malmasi et al., 2022c). The main challenge of the previous edition was to perform the NER task in a small context.

Finally, we decided to use the PolDeepNer2 (Marcińczuk and Radom, 2021) package as the foundation for our research. It is based on a transformer architecture and thus allows the use of pre-trained language models. It differs from traditional token classification models in the way of representation subject to classification – classification is done on tokens, which represent single words, not subwords. In addition to this, it has different methods for adding a context representation to the sentence being analyzed.

## 3 Data

In our research, we used solely the dataset provided by the shared task organizers (Fetahu et al., 2023a; Malmasi et al., 2022a). The dataset was provided in the CoNLL format and was limited to two columns: the first one contains tokens text form and the fourth contains tokens label in the IOB2 format. Samples sentenced are presented in Figure 1.

```
# id 309f5b26-951e-472b-948e-47632249862b domain=en
robert _ _ B-OtherPER
gottschalk _ _ I-OtherPER
1939 _ _ O
academy _ _ B-VisualWork
award _ _ I-VisualWork
winner _ _ O
and _ _ O
founder _ _ O
of _ _ O
panavision _ _ B-ORG


# id bb81b9a7-e73d-4977-b6a8-0f7937123dfe domain=en
during _ _ O
the _ _ O
reign _ _ O
of _ _ O
the _ _ O
tongzhi _ _ B-OtherPER
emperor _ _ I-OtherPER
( _ _ O
r _ _ O
. _ _ O
1861 _ _ O
– _ _ O
1875 _ _ O
) _ _ O
: _ _ O
```

Figure 1: The two first sentences from the English training subset.

## 4 Methodology

Our system employs a pre-trained masked language model and a fully-connected layer performing token head classification. The classification layer features dropout regularization. We used the *mLUKE-large* (Yamada et al., 2020a) model as the underlying pre-trained masked language model. The model is trained solely on the data provided by the organizers for this shared task. We used data augmentation on the level of sequence representation using different strategies of feeding data to the network.

### 4.1 MLM selection

We considered three existing pre-trained models:

- `bert-uncased-large`[1] (Devlin et al., 2018) — this is the only tested model that was trained on uncased texts similar to the shared task data.
- `xlm-roberta-large`[2] (Conneau et al., 2019) — this is a widely used model for multilingual named entity recognition which allows for the SOTA results.
- `mluke-large`[3] (Ri et al., 2022) — this is the `xlm-roberta-large` model fined-tuned with entity representations using Wikipedia for 24 languages. Ri et al. (2022) showed that the fine-tuned model outperformed the base model in the named entity recognition for English by 1.5pp.

| Model | P | R | F |
|---|---|---|---|
| `bert-uncased-large` | 72.42 | 75.39 | 73.88 |
| `xlm-roberta-large` | 69.81 | 71.37 | 70.58 |
| `mluke-large` | 70.44 | 72.07 | 71.24 |

Table 1: Comparision of different MLM models

Table 1 compares the results obtained for the three models for the English development subset using data augmentation presented in this article. The `bert-uncased-large` model got the highest F-measure of 73.88% and outperformed the other two models `xlm-roberta-large` and `mluke-large`. We attribute the better performance

[1] https://huggingface.co/bert-large-uncased
[2] https://huggingface.co/xlm-roberta-large
[3] https://huggingface.co/studio-ousia/mluke-large

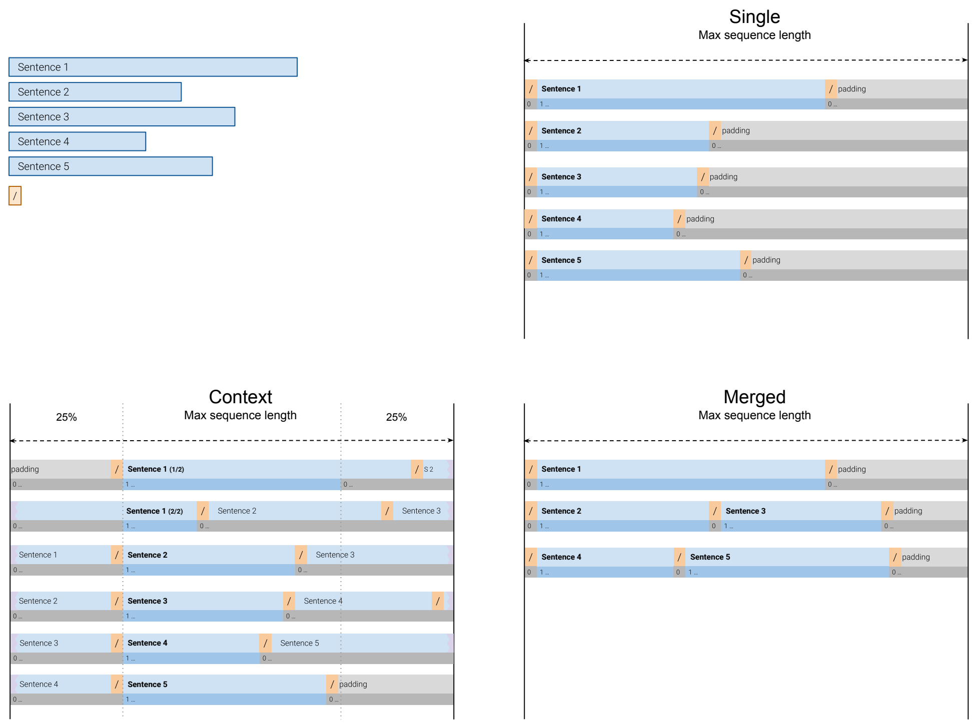


Figure 2: An overview of the sequence generation strategies

to the fact that `bert-uncased-large` was trained on uncased text, while the other two models were trained on cased texts. MultiCoNER II datasets were also uncased, leading to better tokenization and semantic representation. For instance, the first name *christoph* is tokenized by `mluke-large` into a single token, while the other two models split it into two subtokens: *christ* and *oph*.

Although `bert-uncased-large` achieved the highest score, we used the `mluke-large`, which is the second-best model. This is due to a misinterpretation of our initial results, according to which the BERT model performed worse than the other two models.

### 4.2 Token representation

Each token (text form) is tokenized into a sequence of subtokens. We take up to six subtokens for each token. The first subtoken (head) is subjected to classification. The multi-head attention mechanism uses the remaining five tokens to calculate the representation of each head in the sequence.

We decided to trim subtokens to six elements to reduce the impact of the over-tokenized words. In Table 2, we present sample over-tokenized words.

| Token | Subtokens | Count |
|---|---|---|
| s.t.a.l.k.e.r. | s . t. a . l . k . e . r . | 14 |
| 81-717/81-714 -type | 81 - 71 ##7 / 81 - 71 ##4 - type | 11 |
| immaculateconceptioncathedral jf131 | immaculate ##con ##ception ##cat ##hedral ##j ##f ##13 ##1 | 9 |

Table 2: Sample over-tokenized words from the English dataset.

### 4.3 Sequence length

We used the sequence length of 128 subtokens. We decided to use this length based on the distribution of sentences' size in the training dataset. 98% of sentences contained up to 32 subtokens (see Table 3). For 128 subtokens, most vectors could fit up more than four sentences, which is sufficient for our setup.

### 4.4 Data augmentation

We used data augmentation by combining different strategies of sequence generation (named *single*, *merged*, and *context*). The strategies are presented in Figure 2. Each sentence is used three times as

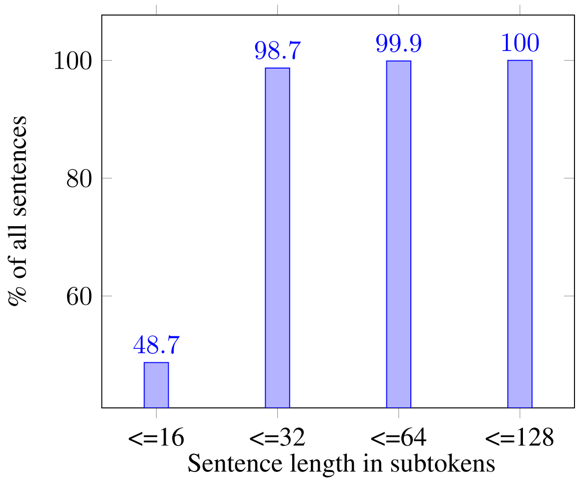


Figure 3: Distribution of sentence lengths in the training subset.

a training example. Each time the input sequence is constructed differently, which affects the vector representation of the token heads. The representations of the token head vary due to the multi-head attention mechanism and differences in the token context.

**Sentence**: christoph haberland designed a new marble pulpit for the church which was built in italy in 1793 .

**Contexts**:

1. eli lilly founder president of pharmaceutical company eli lilly and company
2. he was succeeded as chancellor by sir frank kitto
3. a blue balloon dog sculpture created by Koons broke into tiny shards when a visitor accidentally kicked its podium, according to the gallery hosting the piece .
4. bel-air fine art was displaying the piece at its booth at Art Wynwood, a contemporary art fair in miami .

Figure 4: Sample sentences used to demonstrate similarity range for single subwords based on the context.

In Figure 4, we present a sample sentence and several contexts. We concatenated the context with the sentence and fed it into the pre-trained language model for each context. Then, we took subtoken vector representations and computed the cosine similarity between the corresponding subtokens with and without context. In Table 3, we presented the similarity values for two subtokens: *chritoph* and *italy*. As we can observe, for *chritoph*, the similarity varies from 0.948 to 0.764. This indicates that the subtokens' vectors differ significantly. Based on this observation, we argue that simple sentence concatenation with any sentences can augment training data and improve the performance of the final model.

| **Subtoken** | **Similarity** (descending order) |
|---|---|
| *christoph* | 0.948, 0.876, 0.873, 0.764 |
| *italy* | 0.971, 0.974, 0.954, 0.947 |

Table 3: The similarity between embeddings generated for the same subtokens in different contexts.

We used three strategies of sequence generation:

- *single* — a vector contains a single sentence. In the training dataset, all sentences are shorter than 128 subtokens.
- *merged* — a vector contains that many consecutive sentences as fit into a sequence of 128 subtokens. The sentences are separated with a special subtoken. Each token head in the sequence is subjected to classification.
- *context* — a vector contains up to 64 subtokens subjected to classification, 32 preceding, and 32 following subtokens as the context. The subtokens from the context are used only for embedding calculation by the language model and are not subjected to classification.

### 4.5 Training parameters

During training, we modify the weights of the classification layers and the pre-trained masked model. The models were trained for 20 epochs, with a learning rate decay from $5e-6$, a dropout rate of 0.2, and a batch size of 16.

## 5 Results

In Tables 4 and 5, we present results obtained on the development subsets for English and multilingual datasets, respectively. To verify our hypothesis that data augmentation by simple sentence concatenation with different sentences can improve performance, we trained the models in two setups — *single* and *union*. In the *single* setup we trained the model using single sentences as vectors. In the *union* setup we combined all three strategies, i.e. *single*, *merged*, and *context*.

For English and multilingual datasets, the highest score was obtained for the *union* setup with

| Train | Eval | P | R | F |
|---|---|---|---|---|
| single | single | 68.38 | 70.78 | 69.55 |
| union | single | 70.44 | 72.07 | 71.24 |
| | merged | 74.83 | 76.85 | 75.83 |
| | + shuffle | 65.88 | 66.74 | 66.31 |
| | context | 77.26 | 77.85 | 77.56 |
| | + shuffle | 68.86 | 68.75 | 68.80 |

Table 4: The evaluation results on the English development subset.

| Train | Eval | P | R | F |
|---|---|---|---|---|
| single | single | 75.20 | 77.73 | 76.44 |
| union | single | 79.05 | 79.97 | 79.51 |
| | merged | 82.37 | 83.75 | 83.05 |
| | + shuffle | 71.49 | 72.59 | 72.03 |
| | context | 84.28 | 85.25 | 84.77 |
| | + shuffle | 75.20 | 75.84 | 75.52 |

Table 5: The evaluation results on the multilingual development subset.

*context* representation on inference. However, in the context of MultiCoNER II dataset, the evaluation might be unreliable because the consecutive sentences in the testing dataset might not be related to each other. To simulate this scenario, we shuffled the sentences and processed them in a random order (+*shuffle*). For the *single* strategy on inference, we obtained the same result as for the original order – 71.24% and 79.51% of F-measure, respectively. In the case of both context-aware strategies (*merged* and *context*), the results dropped significantly below the score for the *single* strategy. The drop was ca. 9-10 pp for both datasets and strategies. The drop might indicate that the order of sentences in the development set was not fully randomized. Nevertheless, the most reliable strategy for inference was *single*, as it was not dependent on the context and thus on the order of sentences.

The experiments' results confirmed that combining various strategies for sequence generation during training can improve the model's performance even when using the *single* strategy on inference. For English, when we used *single* strategy for training and inference, we obtained 69.55% of F-measure. For the combination of various strategies (*union*) and *single* on inference we got 71.24%. For the multilingual dataset, we obtained an even greater improvement from 76.44% to 79.51%

On the test datasets, our models got 63.51% and 73.22% of F-measure, respectively, for English and multilingual datasets.

## 6 Conclusion

Our experiments showed that we could improve the performance of a model for named entity recognition using data augmentation on the sequence generation level without any additional data sources. We benefit from training the model with and without context, even when the context was unrelated to the sentence, and on the inference, we processed the sentence separately. The presented data augmentation technique helped improve the F-measure on the English development subset by 1.7 pp and the multilingual dataset by 3.1 pp.

## Acknowledgements

This work was financed by the European Regional Development Fund as a part of the 2014-2020 Smart Growth Operational Programme, CLARIN – Common Language Resources and Technology Infrastructure, project no. POIR.04.02.00-00C002/19.

## References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. 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. *CoRR*, abs/1810.04805.

Besnik Fetahu, Zhiyu Chen, Sudipta Kar, Oleg Rokhlenko, and Shervin Malmasi. 2023a. MultiCoNER v2: a Large Multilingual dataset for Fine-grained and Noisy Named Entity Recognition.

Besnik Fetahu, Sudipta Kar, Zhiyu Chen, Oleg Rokhlenko, and Shervin Malmasi. 2023b. SemEval-2023 Task 2: Fine-grained Multilingual Named Entity Recognition (MultiCoNER 2). In *Proceedings of*

| Language | P | R | F1 | Clean F1 | Noisy F1 | Macro F1 |
|---|---|---|---|---|---|---|
| | Dev | | | Test | | |
| English | 70.44 | 72.07 | 71.24 | 66.04 | 57.84 | 63.51 |
| Multilingual | 79.05 | 79.97 | 79.51 | 73.22 | - | 73.22 |

Table 6: The final evaluation results of our models.

*the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics.

Jinlan Fu, Xuanjing Huang, and Pengfei Liu. 2021. SpanNER: Named entity re-/recognition as span prediction. In *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)*, pages 7183–7195, Online. Association for Computational Linguistics.

Saul A. Kripke. 1980. *Naming and Necessity: Lectures Given to the Princeton University Philosophy Colloquium*. Cambridge, MA: Harvard University Press.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020a. 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. Association for Computational Linguistics.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020b. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Jouni Luoma and Sampo Pyysalo. 2020. Exploring cross-sentence contexts for named entity recognition with bert.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022a. MultiCoNER: A large-scale multilingual dataset for complex named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3798–3809, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022b. SemEval-2022 task 11: Multilingual complex named entity recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1412–1437, Seattle, United States. Association for Computational Linguistics.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022c. SemEval-2022 Task 11: Multilingual Complex Named Entity Recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Michał Marcińczuk and Jarema Radom. 2021. A single-run recognition of nested named entities with transformers. *Procedia Computer Science*, 192:291–297. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 25th International Conference KES2021.

Stephen Mayhew, Gupta Nitish, and Dan Roth. 2020. Robust named entity recognition with truecasing pretraining. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8480–8487.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Ryokan Ri, Ikuya Yamada, and Yoshimasa Tsuruoka. 2022. mLUKE: The power of entity representations in multilingual pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7316–7330, Dublin, Ireland. Association for Computational Linguistics.

Asahi Ushio and Jose Camacho-Collados. 2021. T-ner: An all-round python library for transformer-based named entity recognition. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 53–62.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021a. Automated concatenation of embeddings for structured prediction. In *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)*, pages 2643–2660, Online. Association for Computational Linguistics.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021b. Improving named entity recognition by external context retrieving and cooperative learning. In *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)*, pages 1800–1812, Online. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. Ontonotes release 5.0.

Lu Xu, Zhanming Jie, Wei Lu, and Lidong Bing. 2021. Better feature integration for named entity recognition. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3457–3469, Online. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020a. Luke: Deep contextualized entity representations with entity-aware self-attention. In *EMNLP*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020b. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Deming Ye, Yankai Lin, and Maosong Sun. 2021. Pack together: Entity and relation extraction with levitated marker. *arXiv preprint arXiv:2109.06067*.

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