NLPeople at SemEval-2023 Task 2: A Staged Approach for Multilingual Named Entity Recognition

Mohab Elkaref, Nathan Herr, Shinnosuke Tanaka, Geeth De Mel
IBM Research Europe
Daresbury, United Kingdom
{mohab.elkaref, nathan.herr, shinnosuke.tanaka}@ibm.com
geeth.demel@uk.ibm.com

Abstract

The MultiCoNER II shared task aims at detecting complex, ambiguous named entities with fine-grained types in a low context setting. Previous winning systems incorporated external knowledge bases to retrieve helpful contexts. In our submission we additionally propose splitting the NER task into two stages, a Span Extraction Step, and an Entity Classification step. Our results show that the former does not suffer from the low context setting comparably, and in so leading to a higher overall performance for an external KB-assisted system. We achieve 3^{rd} place on the multilingual track and an average of 6^{th} place overall.

1 Introduction

The 2022 MultiCoNER shared task (Malmasi et al., 2022b,a) aimed at addressing the challenges present in processing semantically-ambiguous and complex Named Entities (NE) in short and low-context settings. The top-performing systems made use of existing knowledge in the form of knowledge-bases (Wang et al., 2022b) and gazeteers (Chen et al., 2022). The organisers of the MultiCoNER task noted, however, that such solutions are not robust against NEs which are not present in the existing knowledge or are noisy[1]. Such findings are the key motivators behind the 2023 MultiCoNER shared task (Fetahu et al., 2023b), with a far more fine-grained set of classes and noise introduced into some additional datasets (Fetahu et al., 2023a).

We propose splitting the standard Named Entity Recognition (NER) task into two stages: (1) Span Extraction and (2) Entity Classification. The reason being that span extraction is closely tied to language and thus the noisiness of the entities or their existence in the knowledge, should not greatly affect its results. Following which, the classification of the extracted spans should be simpler, as the spans will have already been extracted. Furthermore, we make use of different types of additional contexts based off of the Wikipedia dumps for each language (Wang et al., 2022b). The details of which can be found in sub-section 3.2.

2 Related Work

Recent approaches to NER can generally be split into three groups 1. CRF-based, 2. Linear, and 3. Span Enumeration.

CRF-based approaches such as Lample et al. (2016) and Straková et al. (2019) use a bidirectional LSTM with a Conditional Random Field (CRF) as a sequence tagger, relying on one or a combination of word embeddings, character representations, and transformer representations as input features to the combined LSTM-CRF decoder.

Linear approaches such as Luoma and Pyysalo (2020) and Schweter and Akbik (2021) rely on fine-tuning a pretrained transformer architecture with a linear layer to greedily decode the output. The representations passed onto the linear layer can also include word embeddings such as in Schweter and Akbik (2021). As an alternative to the greedy approach, Luoma and Pyysalo (2020) combines scores for each sentence from different positions in the input context.

Finally, approaches such as (Yu et al., 2020; Yamada et al., 2020) classify all possible spans within a sentence. These spans are then ranked by their highest score and each span is included in the final output if it does not overlap with a higher scoring span that is already included in the final output.

Regardless of the method used, most state-of-the-art models use either a large transformer representation and/or a combination of sources for their word embeddings. Straková et al. (2019) use ELMo, BERT, and Flair embeddings, while Lample et al. (2016) use both word and character

[1] Examples of noisy Named Entities are ones with spelling mistakes and typos.

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1148–1153
July 13-14, 2023 ©2023 Association for Computational Linguistics

embeddings, and Yu et al. (2020) use BERT, fastText, and character embeddings. Wang et al. (2020) propose training a controller to automatically concatenate embeddings from different sources based on the current context. Finally, Schweter and Akbik (2021) rely on XLM-R embeddings and word embeddings from different sources depending on the target language. Yamada et al. (2020) on the other hand, train a transformer model that produces contextualised embeddings for words and entities in a sentence, independent of each other. This entity-aware transformer is then tuned on a span enumeration task as described above.

From the 2022 MultiCoNER shared task, the two winning systems used distinct strategies. Wang et al. (2022b) relied on large-scale retrieval of relevant paragraphs to the target sentence, which were then concatenated and used as input to a Transformer-CRF system. Chen et al. (2022) on the other hand use a gazetteer-augmented-BiLSTM model together with a transformer model to classify target sentences. The BiLSTM is pre-trained to produce token embeddings similar to the accompanying transformer when given sequence labels based on gazetteer matches.

3 System Description

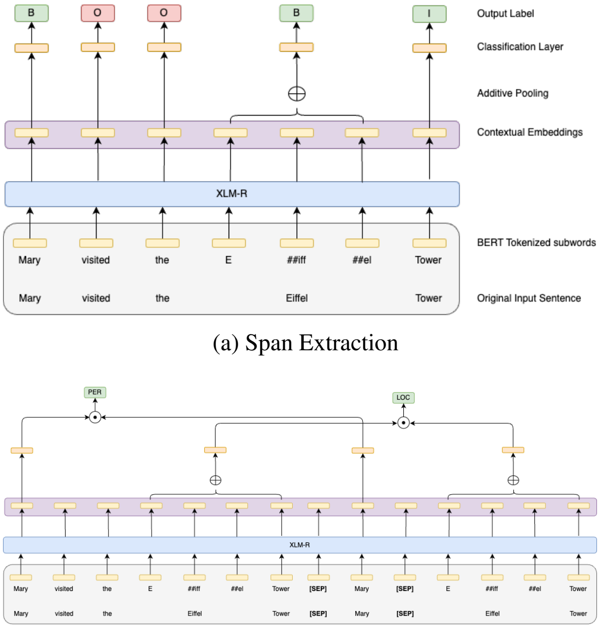
3.1 Staged-NER

Our approach consists of two separate stages, a **Span Extraction** stage and an **Entity Classification** stage.

The Span Extraction stage Our first stage is a conceptually simple variant of a standard transformer fine-tuning task (Figure 1a). We tune a pre-trained transformer on the training data, but with the modified task of classifying a token as one of $C_{\text{Span}} = [\texttt{B,I,O}]$. Thus this model only detects entities without needing to classify *which kind* of entity it is. The token representation passed onto the classification layer is obtained through the additive pooling of the sub-word tokens corresponding to each token. Given a sentence's sub-word representation X, the final contextual representation r_i^t for token t_i is expressed as

$$r_i^t = \Sigma_{j=START^t(i)}^{END^t(i)} x_j$$

where $START^t(i)$ and $END^t(i)$ are the start and end indices of the sub-words constituting token t_i.



(a) Span Extraction

(b) Entity Classification

Figure 1: Illustration of our approach

The Entity Classification stage Given a sentence's sub-word representation X, a set of entity spans $S = \{s_1, s_2, ..., s_m\}$ in X, and a set of entity forms $F = \{f_1, f_2, ..., f_m\}$ the entity typing model computes an entity-aware representation of each span in S (Figure 1b). Similar to the span extraction stage, an entity's encoding is the sum of the contextual representation for each word/sub-word in the entity span *concatenated with* the sum of the representation of each word/sub-word in the corresponding entity form.

$$s_i = \Sigma_{j=START^e(i)}^{END^e(i)} x_j$$

$$f_i = \Sigma_{j=START^f(i)}^{END^f(i)} x_j$$

$$r_i^e = s_i \odot f_i$$

where $START^e(i)$ and $END^e(i)$ are the start and end indices of entity span s_i, and $START^f(i)$ and $END^f(i)$ are the start and end indices of entity form f_i. Each entity representation is then classified using a final output layer.

3.2 Additional Context

In addition to the target sentence and the entity forms appended to the sentence in the Entity Classification stage, we also retrieve external context from Wikipedia. In this section we describe the external context incorporated in various formats during both stages. We explain the process of obtaining and formatting data from Wikipedia in subsection 3.3.

We use BM25 (Robertson et al., 2009) to index the summary paragraphs for all articles in each language's respective Wikipedia.

Matched Titles We extract any strings in the target sentence that exactly match article titles. We found that many languages benefit from string matching against titles from other languages in addition to their own, likely due to code-switching.

Linked Page Titles For the top 15 paragraphs similar to the target sentence, we retrieve the page titles of each page linked-to in these paragraphs, in addition to the titles of the pages in which these paragraphs appear.

Combined We additionally experiment with combining candidate titles from both the *Matched Titles* and *Linked Page Titles* settings.

Paragraphs The summary paragraphs of articles whose titles appear in the target sentence. Similar to *Matched Titles* we compare to titles in all languages, but only use a single paragraph per entity. For entities with paragraphs in multiple languages we prioritise paragraphs in the target language, otherwise we choose the language which has the most articles in its respective Wiki.

For sentences with multiple matched entities we concatenate their summary paragraphs separated by the <sep> token. In order to ensure that the target sentence concatenated with relevant paragraphs does not exceed the sequence limit, we include the first 50 tokens of each paragraph.

Infobox Infobox labels (e.g., Born, Citizenship, Occupation for an article about a person) were obtained from Wikipedia pages. The labels of the article title that matched the span of the given sentence were used as additional context. For the multilingual dataset, all labels were translated into English using OPUS-MT (Tiedemann and Thottingal, 2020).

3.3 WikiData Database

We downloaded the latest wiki-data dump for each language[2]. Once each language's dump had been downloaded and verified, we used WikiExtractor[3] to extract and clean the text and output the results into a readable Json format. Subsequently, a Wiki-Database was constructed. The structure of our database is illustrated in Figure 2.

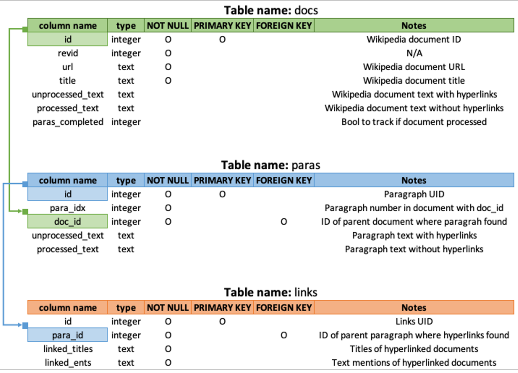


Table name: docs

column name	type	NOT NULL	PRIMARY KEY	FOREIGN KEY	Notes
id	integer	O	O		Wikipedia document ID
revid	integer	O			N/A
url	text	O			Wikipedia document URL
title	text	O			Wikipedia document title
unprocessed_text	text				Wikipedia document text with hyperlinks
processed_text	text				Wikipedia document text without hyperlinks
paras_completed	integer				Bool to track if document processed

Table name: paras

column name	type	NOT NULL	PRIMARY KEY	FOREIGN KEY	Notes
id	integer	O	O		Paragraph UID
para_idx	integer	O			Paragraph number in document with doc_id
doc_id	integer	O		O	ID of parent document where paragrah found
unprocessed_text	text				Paragraph text with hyperlinks
processed_text	text				Paragraph text without hyperlinks

Table name: links

column name	type	NOT NULL	PRIMARY KEY	FOREIGN KEY	Notes
id	integer	O	O		Links UID
para_id	integer	O		O	ID of parent paragraph where hyperlinks found
linked_titles	text	O			Titles of hyperlinked documents
linked_ents	text	O			Text mentions of hyperlinked documents

Figure 2: Figure showing structure of our database constructed from Wikipedia dump.

3.4 BM25 Candidate Retrieval

The first paragraphs of Wikipedia in each language was used as a corpus, and the similarity to the given sentences was calculated using BM25. The top 15 similar paragraphs were retrieved from the corpus and used as the candidates. For the multilingual dataset, we combined the first paragraph of Wikipedia in all 12 languages and used it as the corpus. Our indexing parameters for BM25 were k_1=1.5 and b=0.75.

3.5 Final Model

For each stage we train models for different context types. For each context setting we train 5 models with different initial random seeds and pool the resulting scores. While the contexts types listed in Section 3.2 allow for many combinations, we experimented with a limited set due to time and computational constraints.

Span Extraction Ensemble we train three variants with three context types for each language; (1) *Matched Titles*, (2) *Linked Page Titles*, and (3) *Combined*.

Entity Classification Ensemble we train models for four context types for each language; (1) *Matched Titles*, (2) *Linked Page Titles*, (3) *Paragraphs*, and (4) *Infobox*.

[2] https://dumps.wikimedia.org/*wiki/latest/*wiki-latest-pages-articles.xml.bz2 where * is one of {bn, de, en, es, fa, fr, hi, it, pt, sv, uk, or zh}

[3] https://github.com/attardi/wikiextractor/wiki

Our final model for each stage is an ensemble of all the trained models above, with the final prediction obtained through voting by all models. The Entity Classification models were trained on predicted spans produced by the Span Extraction Ensemble for each language on both the train and development sets.

4 Experimental Setup

Implementation Details We implement our models in Pytorch and use pretrained models provided through the HuggingFace transformers library. We demonstrate our approach using XLM-Roberta(XLM-R)[4] (Conneau et al., 2020).

Training & Hyperparameters We ran a limited grid search similar to Luoma and Pyysalo (2020) to obtain the best hyperparameters for each model. The settings explored were learning rate $\in \{5 \times 10^{-6}, 2 \times 10^{-5}\}$, batch size $\in \{4, 16\}$. The loss function used for both stage models was a softmax cross-entropy.

$$p(r_c^*) = \frac{\exp(\text{Score}(r_c^*))}{\sum_{\hat{c}=1}^{C} \exp(\text{Score}(r_{\hat{c}}^*))}$$

$$loss = -\sum_{i=1}^{N} \sum_{c=1}^{C} y_{i_c} \log(p(r_c^*))$$

where r^* is r^t for span extraction and baseline models, r^e for entity classification models, $C \in \{C_{\text{Span}}, C_{\text{Entity}}\}$ depending on the task, y is the expected output, and N is the batch size. We ran all our experiments for 5 epochs with a linear decay schedule and no warm-up.

5 Results and Analysis

We present the results of our approach on the development sets in Tables 1 & 2. In order to get an accurate view of how well each stage is performing, we calculate F1 scores both with and without tags. We call this *Labelled F1* and *Unlabelled F1* respectively. In both cases we calculate span F1, where an entity is correct if the start and end of the span are correct, and for labelled F1 the assigned tag must also be correct. Following the organisers we present macro-F1 scores for our labelled F1 results.

The results of the Span Extraction stage is shown in Table 1. As we hypothesised, span extraction is comparably easier than entity classification, with our model achieving an unlabelled F1 score above 90% on 11 of the 13 languages. This holds for both the pooled model, and all three variants with different contexts.

Additionally, while Matched Title and Liked Page Title contexts don't clearly outperform each other across languages, the Combined context beats both on 11 of the 13 languages, with a negligible drop in performance on the remaining two.

The results of our Entity Classification stage are shown in Table 2. For this stage we compare our results to two baselines provided by the authors of the winning system of last year's MultiCoNER shared task (Malmasi et al., 2022b,a). The first baseline is a simple tuned XLM-R model with a CRF Layer. The second baseline, RaNER, is a variant of this same model that utilises the retrieval augmented approach used by the winning team of the last shared task (Wang et al., 2022a).

Our models outperform the weaker CRF baseline across the board for all languages, including all individual models, with the exception of our Matched Titles model for Hindi. Additionally, our final pooled models outperformed the stronger RaNER baseline on 7 of 12 languages (multilingual baseline scores were not available).

While the Matched Title and Linked Page Title models beat the RaNER baseline on 6 of 12 languages, they are generally outperformed by the Infobox and Paragraphs models. The improved performance of the Infobox model is a result of the class specific properties present in the Infobox of the matched entity. For example *Albert Einstein's* Infobox has keys such as *Born*, *Citizenship*, and *Education* which are all clearly properties of a person and thus make labelling *Albert Einstein* as person a simpler task. Similarly, the improved performance of the Paragraph model is a consequence of the more complete context provided by the first paragraph of the matched entity. For example, the wikipedia page of *J. Robert Oppenheimer* says that they are *a professor of physics at the University of California, Berkeley* where *physics* is a hyperlink and will be present in the additional contexts for Matched Title and Linked Page Title models but what is actually important in that sentence is that they are a *professor of physics*, indicating that they are a scientist.

Paragraph-based models were generally the best performing models, beating the other contexts on 9

[4] https://huggingface.co/xlm-roberta-large

Context Type	bn	de	en	es	fa	fr	hi	it	pt	sv	uk	zh	multi
MT	96.25	94.24	92.44	90.41	**83.17**	91.71	95.77	93.55	91.66	93.28	88.60	92.54	92.45
LPT	94.48	93.70	93.29	91.27	80.45	91.54	92.03	94.10	**92.70**	92.95	88.37	90.52	92.04
Combined	95.96	94.68	93.85	**91.51**	83.10	91.89	**96.50**	94.27	92.55	93.91	88.94	92.51	92.78
Pooled	**96.47**	**95.09**	**94.01**	91.35	82.93	**92.56**	95.91	**94.42**	92.52	**93.96**	**89.05**	**92.88**	**93.11**

Table 1: Unlabelled F1 results on the dev. sets for the Span Extraction stage. MT is Matched Titles, LPT is Linked Page Titles. Best results are in **Bold**, and second-best results are underlined.

Context Type	bn	de	en	es	fa	fr	hi	it	pt	sv	uk	zh	multi
Baselines													
CRF	77.06	73.17	60.68	65.04	59.40	61.41	83.80	71.12	63.94	68.40	65.71	72.60	-
RaNER	**89.12**	76.78	71.32	68.24	**76.76**	74.61	**88.78**	**83.43**	76.70	77.06	**78.26**	75.84	-
This Work													
MT	85.69	78.75	73.40	77.04	71.50	77.05	83.76	76.39	76.60	79.67	73.36	75.85	-
LPT	84.60	80.06	74.54	76.84	72.38	77.30	84.49	77.21	75.73	78.14	73.73	72.43	81.77
Infobox	87.55	80.28	76.46	78.05	73.26	77.32	86.98	77.07	77.69	78.65	74.75	76.18	83.38
Paragraphs	87.40	80.71	76.07	78.55	72.42	78.15	87.08	78.31	78.30	80.17	74.86	74.87	**85.13**
Pooled	88.45	**81.93**	**77.26**	**80.14**	73.84	**79.26**	88.31	79.01	**78.78**	**80.98**	75.66	**77.45**	84.16

Table 2: Labelled Macro F1 results on the dev. sets for the Entity Classification stage. MT is Matched Titles, LPT is Linked Page Titles. Best results are in **Bold**, and second-best results are underlined.

Lang	Clean		Noisy		Overall		Rank
	unlabelled	labelled	unlabelled	labelled	unlabelled	labelled	
bn	94.51	78.24	-	-	94.51	78.24	5
de	95.42	77.67	-	-	95.42	77.67	5
en	92.23	76.0	84.10	62.23	89.79	71.81	9
es	91.88	77.22	84.36	63.53	89.62	72.76	6
fa	84.55	70.76	-	-	84.55	70.76	4
fr	91.68	77.12	83.76	63.40	89.30	72.85	8
hi	94.38	78.50	-	-	94.38	78.50	5
it	93.91	77.45	87.90	65.88	92.10	73.71	8
pt	92.30	74.50	86.45	62.22	90.55	70.16	8
sv	94.52	79.31	89.59	67.15	93.04	75.08	6
uk	89.60	73.41	-	-	89.60	73.41	5
zh	91.40	71.43	78.53	48.95	87.75	65.96	7
multi	92.39	81.02	87.32	68.32	91.30	78.38	3

Table 3: Final F1 scores on the test set.

of 13 tracks, with Infobox achieving the best results on only 4 languages.

Our final results are shown in Table 3. The performance of our Span Extraction stage remains stable across the dev. and test sets, achieving over 90% unlabelled F1 on 11 of 13 languages on the clean dataset, and remains above 90% on 4 of the 8 languages with additional noisy data. The noisy data in isolation predictably harms performance by 5-8% F1, except for Chinese where performance falls by 13%.

The Entity Classification step also generalised fairly well from the dev. set to the clean test set, with macro F1 scores being 1-6% lower than on the dev. set, with the exception of Bengali and Hindi which fell by 10%. Our approach is strongly impacted by the introduced noise for all languages.

The contrast between our results on noisy and clean data points to the importance of accurate Span Extraction, which acts as an upper bound to the performance of the Entity Classification step. Additionally, the results of Span Extraction were less impacted overall by the introduction of noise than Entity Classification. This is possibly due to the reliance of our Entity Classification on entity matching, while Span Extraction relies primarily on linguistic patterns, with additional context being supplementary.

6 Conclusion

In this work we demonstrated a staged approach to NER, where we learn to extract spans before classifying entities. Our approach achieves strong results on Span Extraction and is resilient to noise, while the Entity Classification stage is far more sensitive and dependant on the retrieved context. We achieve 3^{rd} place on the multilingual track and an average of 6^{th} place overall.

References

Beiduo Chen, Jun-Yu Ma, Jiajun Qi, Wu Guo, Zhen-Hua Ling, and Quan Liu. 2022. Ustc-nelslip at semeval-2022 task 11: Gazetteer-adapted integration network for multilingual complex named entity recognition. *arXiv preprint arXiv:2203.03216*.

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. Association for Computational Linguistics.

Besnik Fetahu, Zhiyu Chen, Sudipta Kar, Oleg Rokhlenko, and Shervin Malmasi. 2023a. Multi-CoNER 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 the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

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

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.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Stefan Schweter and Alan Akbik. 2021. Flert: Document-level features for named entity recognition.

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. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2020. Automated concatenation of embeddings for structured prediction. *arXiv preprint arXiv:2010.05006*.

Xinyu Wang, Yongliang Shen, Jiong Cai, Tao Wang, Xiaobin Wang, Pengjun Xie, Fei Huang, Weiming Lu, Yueting Zhuang, Kewei Tu, Wei Lu, and Yong Jiang. 2022a. DAMO-NLP at SemEval-2022 task 11: A knowledge-based system for multilingual named entity recognition. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1457–1468, Seattle, United States. Association for Computational Linguistics.

Xinyu Wang, Yongliang Shen, Jiong Cai, Tao Wang, Xiaobin Wang, Pengjun Xie, Fei Huang, Weiming Lu, Yueting Zhuang, Kewei Tu, et al. 2022b. Damo-nlp at semeval-2022 task 11: A knowledge-based system for multilingual named entity recognition. *arXiv preprint arXiv:2203.00545*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. 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.

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.