Named Entity Recognition as Dependency Parsing

Juntao Yu
Queen Mary University
London, UK
juntao.yu@qmul.ac.uk

Bernd Bohnet
Google Research
Netherlands
bohnetbd@google.com

Massimo Poesio
Queen Mary University
London, UK
m.poesio@qmul.ac.uk

Abstract

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing, concerned with identifying spans of text expressing references to entities. NER research is often focused on flat entities only (flat NER), ignoring the fact that entity references can be nested, as in *[Bank of [China]]* (Finkel and Manning, 2009). In this paper, we use ideas from graph-based dependency parsing to provide our model a global view on the input via a biaffine model (Dozat and Manning, 2017). The biaffine model scores pairs of start and end tokens in a sentence which we use to explore all spans, so that the model is able to predict named entities accurately. We show that the model works well for both nested and flat NER through evaluation on 8 corpora and achieving SoTA performance on all of them, with accuracy gains of up to 2.2 percentage points.

1 Introduction

'Nested Entities' are named entities containing references to other named entities as in *[Bank of [China]]*, in which both *[China]* and *[Bank of China]* are named entities. Such nested entities are frequent in data sets like ACE 2004, ACE 2005 and GENIA (e.g., 17% of NEs in GENIA are nested (Finkel and Manning, 2009), altough the more widely used set such as CONLL 2002, 2003 and ONTONOTES only contain so called flat named entities and nested entities are ignored.

The current SoTA models all adopt a neural network architecture without hand-crafted features, which makes them more adaptable to different tasks, languages and domains (Lample et al., 2016; Chiu and Nichols, 2016; Peters et al., 2018; Devlin et al., 2019; Ju et al., 2018; Sohrab and Miwa, 2018; Straková et al., 2019). In this paper, we introduce a method to handle both types of NEs in one system by adopting ideas from the biaffine dependency parsing model of Dozat and Manning (2017). For dependency parsing, the system predicts a head for each token and assigns a relation to the head-child pairs. In this work, we reformulate NER as the task of identifying start and end indices, as well as assigning a category to the span defined by these pairs. Our system uses a biaffine model on top of a multi-layer BiLSTM to assign scores to all possible spans in a sentence. After that, instead of building dependency trees, we rank the candidate spans by their scores and return the top-ranked spans that comply with constraints for flat or nested NER. We evaluated our system on three nested NER benchmarks (ACE 2004, ACE 2005, GENIA) and five flat NER corpora (CONLL 2002 (Dutch, Spanish) CONLL 2003 (English, German), and ONTONOTES). The results show that our system achieved SoTA results on all three nested NER corpora, and on all five flat NER corpora with substantial gains of up to 2.2% absolute percentage points compared to the previous SoTA. We provide the code as open source[1].

2 Related Work

Flat Named Entity Recognition. The majority of flat NER models are based on a sequence labelling approach. Collobert et al. (2011) introduced a neural NER model that uses CNNs to encode tokens combined with a CRF layer for the classification. Many other neural systems followed this approach but used instead LSTMs to encode the input and a CRF for the prediction (Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016). These latter models were later extended to use context-dependent embeddings such as ELMo (Peters et al., 2018). Clark et al. (2018) quite successfully used cross-view training (CVT) paired with multi-task learning. This method yields impressive gains for

[1]The code is available at https://github.com/juntaoy/biaffine-ner

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6470–6476

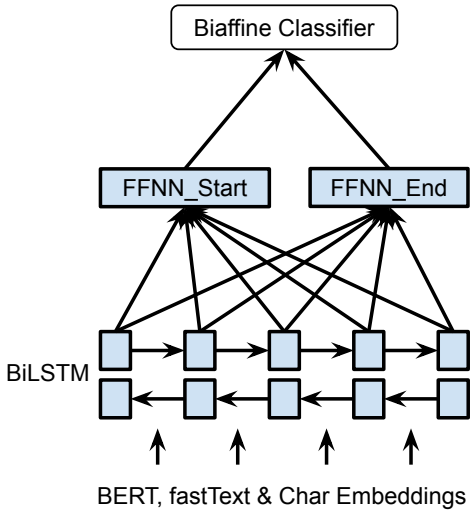

Figure 1: The network architectures of our system.

a number of NLP applications including NER. Devlin et al. (2019) invented BERT, a bidirectional transformer architecture for the training of language models. BERT and its siblings provided better language models that turned again into higher scores for NER.

Lample et al. (2016) cast NER as transition-based dependency parsing using a Stack-LSTM. They compare with a LSTM-CRF model which turns out to be a very strong baseline. Their transition-based system uses two transitions (shift and reduce) to mark the named entities and handles flat NER while our system has been designed to handle both nested and flat entities.

Nested Named Entity Recognition. Early work on nested NER, motivated particularly by the GENIA corpus, includes (Shen et al., 2003; Beatrice Alex and Grover, 2007; Finkel and Manning, 2009). Finkel and Manning (2009) also proposed a constituency parsing-based approach. In the last years, we saw an increasing number of neural models targeting nested NER as well. Ju et al. (2018) suggested a LSTM-CRF model to predict nested named entities. Their algorithm iteratively continues until no further entities are predicted. Lin et al. (2019) tackle the problem in two steps: they first detect the entity head, and then they infer the entity boundaries as well as the category of the named entity. Straková et al. (2019) tag the nested named entity by a sequence-to-sequence model exploring combinations of context-based embeddings such as ELMo, BERT, and Flair. Zheng et al. (2019) use a boundary aware network to solve the nested NER. Similar to our work, Sohrab and Miwa (2018) enumerate exhaustively all possible spans up to a defined length by concatenating the LSTMs outputs for the start and end position and then using this to calculate a score for each span. Apart from the different network and word embedding configurations, the main difference between their model and ours is there for the use of biaffine model. Due to the biaffine model, we get a global view of the sentence while Sohrab and Miwa (2018) concatenates the output of the LSTMs of possible start and end positions up to a distinct length. Dozat and Manning (2017) demonstrated that the biaffine mapping performs significantly better than just the concatenation of pairs of LSTM outputs.

3 Methods

Our model is inspired by the dependency parsing model of Dozat and Manning (2017). We use both word embeddings and character embeddings as input, and feed the output into a BiLSTM and finally to a biaffine classifier.

Figure 1 shows an overview of the architecture. To encode words, we use both BERT$_{Large}$ and fastText embeddings (Bojanowski et al., 2016). For BERT we follow the recipe of (Kantor and Globerson, 2019) to obtain the context dependent embeddings for a target token with 64 surrounding tokens each side. For the character-based word embeddings, we use a CNN to encode the characters of the tokens. The concatenation of the word and character-based word embeddings is feed into a BiLSTM to obtain the word representations (x).

After obtaining the word representations from the BiLSTM, we apply two separate FFNNs to create different representations (h_s/h_e) for the start/end of the spans. Using different representations for the start/end of the spans allow the system to learn to identify the start/end of the spans separately. This improves accuracy compared to the model which directly uses the outputs of the LSTM since the context of the start and end of the entity are different. Finally, we employ a biaffine model over the sentence to create a $l \times l \times c$ scoring tensor (r_m), where l is the length of the sentence and c is the number of NER categories + 1(for non-entity). We compute the score for a span i by:

$$
\begin{aligned}
h_s(i) &= \text{FFNN}_s(x_{s_i}) \\
h_e(i) &= \text{FFNN}_e(x_{e_i}) \\
r_m(i) &= h_s(i)^\top \mathbf{U}_m h_e(i) \\
&\quad + W_m(h_s(i) \oplus h_e(i)) + b_m
\end{aligned}
$$

where s_i and e_i are the start and end indices of the span i, U_m is a $d \times c \times d$ tensor, W_m is a $2d \times c$ matrix and b_m is the bias.

The tensor r_m provides scores for all possible spans that could constitute a named entity under the constrain that $s_i \leq e_i$ (the start of entity is before its end). We assign each span a NER category y':

$$y'(i) = \arg\max \ r_m(i)$$

We then rank all the spans that have a category other than "non-entity" by their category scores ($r_m(i_{y'})$) in descending order and apply following post-processing constraints: For nested NER, a entity is selected as long as it does not *clash* the boundaries of higher ranked entities. We denote a entity i to *clash* boundaries with another entity j if $s_i < s_j \leq e_i < e_j$ or $s_j < s_i \leq e_j < e_i$, e.g. in *the Bank of China*, the entity *the Bank of* clashes boundary with the entity *Bank of China*, hence only the span with the higher category score will be selected. For flat NER, we apply one more constraint, in which any entity containing or is inside an entity ranked before it will not be selected. The learning objective of our named entity recognizer is to assign a correct category (including the non-entity) to each valid span. Hence it is a multi-class classification problem and we optimise our models with softmax cross-entropy:

$$p_m(i_c) = \frac{exp(r_m(i_c))}{\sum_{\hat{c}=1}^{C} exp(r_m(i_{\hat{c}}))}$$

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

4 Experiments

Data Set. We evaluate our system on both nested and flat NER, for the nested NER task, we use the ACE 2004[2], ACE 2005[3], and GENIA (Kim et al., 2003) corpora; for flat NER, we test our system on the CONLL 2002 (Tjong Kim Sang, 2002), CONLL 2003 (Tjong Kim Sang and De Meulder, 2003) and ONTONOTES[4] corpora.

For ACE 2004, ACE 2005 we follow the same settings of Lu and Roth (2015) and Muis and Lu (2017) to split the data into 80%,10%,10% for train, development and test set respectively. To make a fair comparson we also used the same documents as in Lu and Roth (2015) for each split.

For GENIA, we use the GENIA v3.0.2 corpus. We preprocess the dataset following the same settings of Finkel and Manning (2009) and Lu and Roth (2015) and use 90%/10% train/test split. For this evaluation, since we do not have a development set, we train our system on 50 epochs and evaluate on the final model.

For CONLL 2002 and CONLL 2003, we evaluate on all four languages (English, German, Dutch and Spanish). We follow Lample et al. (2016) to train our system on the concatenation of the train and development set.

For ONTONOTES, we evaluate on the English corpus and follow Strubell et al. (2017) to use the same train, development and test split as used in CoNLL 2012 shared task for coreference resolution (Pradhan et al., 2012).

Evaluation Metric. We report recall, precision and F1 scores for all evaluations. The named entity is considered correct when both boundary and category are predicted correctly.

Hyperparameters We use a unified setting for all of the experiments, Table 1 shows hyperparameters for our system.

Parameter	Value
BiLSTM size	200
BiLSTM layer	3
BiLSTM dropout	0.4
FFNN size	150
FFNN dropout	0.2
BERT size	1024
BERT layer	last 4
fastText embedding size	300
Char CNN size	50
Char CNN filter widths	[3,4,5]
Char embedding size	8
Embeddings dropout	0.5
Optimiser	Adam
learning rate	1e-3

Table 1: Major hyperparameters for our models.

[2]https://catalog.ldc.upenn.edu/LDC2005T09
[3]https://catalog.ldc.upenn.edu/LDC2006T06
[4]https://catalog.ldc.upenn.edu/LDC2013T19

[5]In Sohrab and Miwa (2018), the last 10% of the training set is used as a development set, we include their result mainly because their system is similar to ours.

[6]The revised version is provided by the shared task organiser in 2006 with more consistent annotations. We confirmed with the author of Akbik et al. (2018) that they used the revised version.

Model	P	R	F1
ACE 2004			
Katiyar and Cardie (2018)	73.6	71.8	72.7
Wang et al. (2018)	-	-	73.3
Wang and Lu (2018)	78.0	72.4	75.1
Straková et al. (2019)	-	-	84.4
Luan et al. (2019)	-	-	84.7
Our model	87.3	86.0	**86.7**
ACE 2005			
Katiyar and Cardie (2018)	70.6	70.4	70.5
Wang et al. (2018)	-	-	73.0
Wang and Lu (2018)	76.8	72.3	74.5
Lin et al. (2019)	76.2	73.6	74.9
Fisher and Vlachos (2019)	82.7	82.1	82.4
Luan et al. (2019)	-	-	82.9
Straková et al. (2019)	-	-	84.3
Our model	85.2	85.6	**85.4**
GENIA			
Katiyar and Cardie (2018)	79.8	68.2	73.6
Wang et al. (2018)	-	-	73.9
Ju et al. (2018)	78.5	71.3	74.7
Wang and Lu (2018)	77.0	73.3	75.1
Sohrab and Miwa (2018)[5]	93.2	64.0	77.1
Lin et al. (2019)	75.8	73.9	74.8
Luan et al. (2019)	-	-	76.2
Straková et al. (2019)	-	-	78.3
Our model	81.8	79.3	**80.5**

Table 2: State of the art comparison on ACE 2004, ACE 2005 and GENIA corpora for nested NER.

Model	P	R	F1
ONTONOTES			
Chiu and Nichols (2016)	86.0	86.5	86.3
Strubell et al. (2017)	-	-	86.8
Clark et al. (2018)	-	-	88.8
Fisher and Vlachos (2019)	-	-	89.2
Our model	91.1	91.5	**91.3**
CONLL 2003 English			
Chiu and Nichols (2016)	91.4	91.9	91.6
Lample et al. (2016)	-	-	90.9
Strubell et al. (2017)	-	-	90.7
Devlin et al. (2019)	-	-	92.8
Straková et al. (2019)	-	-	93.4
Our model	93.7	93.3	**93.5**
CONLL 2003 German			
Lample et al. (2016)	-	-	78.8
Straková et al. (2019)	-	-	85.1
Our model	88.3	84.6	**86.4**
CONLL 2003 German revised[6]			
Akbik et al. (2018)	-	-	88.3
Our model	92.4	88.2	**90.3**
CONLL 2002 Spanish			
Lample et al. (2016)	-	-	85.8
Straková et al. (2019)	-	-	88.8
Our model	90.6	90.0	**90.3**
CONLL 2002 Dutch			
Lample et al. (2016)	-	-	81.7
Akbik et al. (2019)	-	-	90.4
Straková et al. (2019)	-	-	92.7
Our model	94.5	92.8	**93.7**

Table 3: State of the art comparison on CONLL 2002, CONLL 2003, ONTONOTES corpora for flat NER.

5 Results on Nested NER

Using the constraints for nested NER, we first evaluate our system on nested named entity corpora: ACE 2004, ACE 2005 and GENIA. Table 2 shows the results. Both ACE 2004 and ACE 2005 contain 7 NER categories and have a relatively high ratio of nested entities (about 1/3 of then named entities are nested). Our results outperform the previous SoTA system by 2% (ACE 2004) and 1.1% (ACE 2005), respectively. GENIA differs from ACE 2004 and ACE 2005 and uses five medical categories such as DNA or RNA. For the GENIA corpus our system achieved an F1 score of 80.5% and improved the SoTA by 2.2% absolute. Our hypothesise is that for GENIA the high accuracy gain is due to our structural prediction approach and that sequence-to-sequence models rely more on the language model embeddings which are less informative for categories such as DNA, RNA. Our system achieved SoTA results on all three corpora for nested NER and demonstrates well the advantages of a structural prediction over sequence labelling approach.

6 Results on Flat NER

We evaluate our system on five corpora for flat NER (CONLL 2002 (Dutch, Spanish), CONLL 2003 (English, German) and ONTONOTES. Unlike most of the systems that treat flat NER as a sequence labelling task, our system predicts named entities by considering all possible spans and ranking them. The ONTONOTES corpus consists of documents form 7 different domains and is annotated with 18

	F1	Δ
Our model	89.9	
- biaffine	89.1	0.8
- BERT emb	87.5	2.4
- fastText emb	89.5	0.4
- Char emb	89.8	0.1

Table 4: The comparison between our full model and ablated models on ONTONOTES development set.

fine-grained named entity categories. To predict named entities for this corpus is more difficult than for CONLL 2002 and CONLL 2003. These corpora use coarse-grained named entity categories (only 4 categories). The sequence-to-sequence models usually perform better on the CONLL 2003 English corpus (see Table 3), e.g. the system of Chiu and Nichols (2016); Strubell et al. (2017). In contrast, our system is less sensitive to the domain and the granularity of the categories. As shown in Table 3, our system achieved an F1 score of 91.3% on the ONTONOTES corpus and is very close to our system performance on the CONLL 2003 corpus (93.5%). On the multi-lingual data, our system achieved F1 scores of 86.4% for German, 90.3% for Spanish and 93.5% for Dutch. Our system outperforms the previous SoTA results by large margin of 2.1%, 1.5%, 1.3% and 1% on ONTONOTES, Spanish, German and Dutch corpora respectively and is slightly better than the SoTA on English data set. In addition, we also tested our system on the revised version of German data to compare with the model by Akbik et al. (2018), our system again achieved a substantial gain of 2% when compared with their system.

7 Ablation Study

To evaluate the contribution of individual components of our system, we further remove selected components and use ONTONOTES for evaluation (see Table 4). We choose ONTONOTES for our ablation study as it is the largest corpus.

Biaffine Classifier We replace the biaffine mapping with a CRF layer and convert our system into a sequence labelling model. The CRF layer is frequently used in models for flat NER, e.g. (Lample et al., 2016). When we replace the biaffine model of our system with a CRF layer, the performance drops by 0.8 percentage points (Table 4). The large performance difference shows the benefit of adding a biaffine model and confirms our hypothesis that the dependency parsing framework is an important factor for the high accuracy of our system.

Contextual Embeddings We ablate BERT embeddings and as expected, after removing BERT embeddings, the system performance drops by a large number of 2.4 percentage points (see Table 4). This shows that BERT embeddings are one of the most important factors for the accuracy.

Context Independent Embeddings We remove the context-independent fastText embedding from our system. The context-independent embedding contributes 0.4% towards the score of our full system (Table 4). Which suggests that even with the BERT embeddings enabled, the context-independent embeddings can still make quite noticeable improvement to a system.

Character Embeddings Finally, we remove the character embeddings. As we can see from Table 4, the impact of character embeddings is quite small. One explanation would be that English is not a morphologically rich language hence does not benefit largely from character-level information and the BERT embeddings itself are based on word pieces that already capture some character-level information.

Overall, the biaffine mapping and the BERT embedding together contributed most to the high accuracy of our system.

8 Conclusion

In this paper, we reformulate NER as a structured prediction task and adopted a SoTA dependency parsing approach for nested and flat NER. Our system uses contextual embeddings as input to a multi-layer BiLSTM. We employ a biaffine model to assign scores for all spans in a sentence. Further constraints are used to predict nested or flat named entities. We evaluated our system on eight named entity corpora. The results show that our system achieves SoTA on all of the eight corpora. We demonstrate that advanced structured prediction techniques lead to substantial improvements for both nested and flat NER.

Acknowledgments

This research was supported in part by the DALI project, ERC Grant 695662.

References

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. Association for Computational Linguistics.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Barry Haddow Beatrice Alex and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *Proc. of BioNLP*, pages 65–72.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Timothy Dozat and Christopher Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*.

Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore. Association for Computational Linguistics.

Joseph Fisher and Andreas Vlachos. 2019. Merge and label: A novel neural network architecture for nested NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5840–5850, Florence, Italy. Association for Computational Linguistics.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459, New Orleans, Louisiana. Association for Computational Linguistics.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677, Florence, Italy. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl$_1$) : $i180 - -i182$.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192, Florence, Italy. Association for Computational Linguistics.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867, Lisbon, Portugal. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. 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 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Aldrian Obaja Muis and Wei Lu. 2017. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2608–2618, Copenhagen, Denmark. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*, Jeju, Korea.

Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2003. Effective adaptation of a Hidden Markov Model-based Named Entity Recognizer for the biomedical domain. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

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.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark. Association for Computational Linguistics.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 204–214, Brussels, Belgium. Association for Computational Linguistics.

Bailin Wang, Wei Lu, Yu Wang, and Hongxia Jin. 2018. A neural transition-based model for nested mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1011–1017, Brussels, Belgium. Association for Computational Linguistics.

Changmeng Zheng, Yi Cai, Jingyun Xu, Ho-fung Leung, and Guandong Xu. 2019. A boundary-aware neural model for nested named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 357–366, Hong Kong, China. Association for Computational Linguistics.