Explicitly Capturing Relations between Entity Mentions via Graph Neural Networks for Domain-specific Named Entity Recognition

Pei Chen[1] **Haibo Ding**[2] **Jun Araki**[2] **Ruihong Huang**[1]
[1] Department of Computer Science and Engineering, Texas A&M University
[2] Bosch Research North America
{chenpei, huangrh}@tamu.edu
{Haibo.Ding, Jun.Araki}@us.bosch.com

Abstract

Named entity recognition (NER) is well studied for the general domain, and recent systems have achieved human-level performance for identifying common entity types. However, the NER performance is still moderate for specialized domains that tend to feature complicated contexts and jargonistic entity types. To address these challenges, we propose explicitly connecting entity mentions based on both global coreference relations and local dependency relations for building better entity mention representations. In our experiments, we incorporate entity mention relations by Graph Neural Networks and show that our system noticeably improves the NER performance on two datasets from different domains. We further show that the proposed lightweight system can effectively elevate the NER performance to a higher level even when only a tiny amount of labeled data is available, which is desirable for domain-specific NER.[1]

1 Introduction

Named entity recognition (NER) has been well studied for the general domain, and recent systems have achieved close to human-level performance for identifying a small number of common NER types, such as *Person* and *Organization*, mainly benefiting from the use of Neural Network models (Ma and Hovy, 2016; Yang and Zhang, 2018) and pretrained Language Models (LMs) (Akbik et al., 2018; Devlin et al., 2019). However, the performance is still moderate for specialized domains that tend to feature diverse and complicated contexts as well as a richer set of semantically related entity types (e.g., *Cell*, *Tissue*, *Organ* etc. for the biomedical domain). With these challenges in view, we hypothesize that being aware of the re-occurrences of the same entity as well as semantically related entities will lead to better NER performance for specific domains.

Therefore, we propose to explicitly connect entity mentions in a document that are coreferential or in a tight semantic relation to better learn entity mention representations. Precisely, as shown in Figure 1, we first connect repeated mentions of the same entity even if they are sentences away. For example, the named entity "tumor vasculature" appears both in the *Title* and sentence *S6* but in quite different contexts. Connecting the repeated mentions in a document enables the integration of contextual cues as well as enables consistent predictions of their entity types.

Second, we also connect entity mentions based on sentence-level dependency relations to effectively identify semantically related entities. For example, the two entities in sentence *S3*, "bone marrow" of the type *Multi-tissue Structure* and "endothelial progenitors" of the type *Cell*, are the subject and object of the predicate "contains" respectively in the dependency tree. If the system can reliably predict the type of one entity, we can infer the type of the other entity more easily, knowing that they are closely related on the dependency tree.

We incorporate both relations by using Graph Neural Networks (GNNs), specifically, we use the Graph Attention Networks (GATs) (Velickovic et al., 2018) that have been shown effective for a range of tasks (Sui et al., 2019; Linmei et al., 2019). Empirical results show that our lightweight method can learn better word representations for sequence tagging models and further improve the NER performance over strong LMs-based baselines on two datasets, the AnatEM (Pyysalo and Ananiadou, 2014) dataset from the biomedical domain and the Mars (Wagstaff et al., 2018) dataset from the planetary science domain. In addition, considering the lack of annotations challenge for

[1]The code for the system is available here: https://github.com/brickee/EnRel-G

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 735–742
August 1–6, 2021.

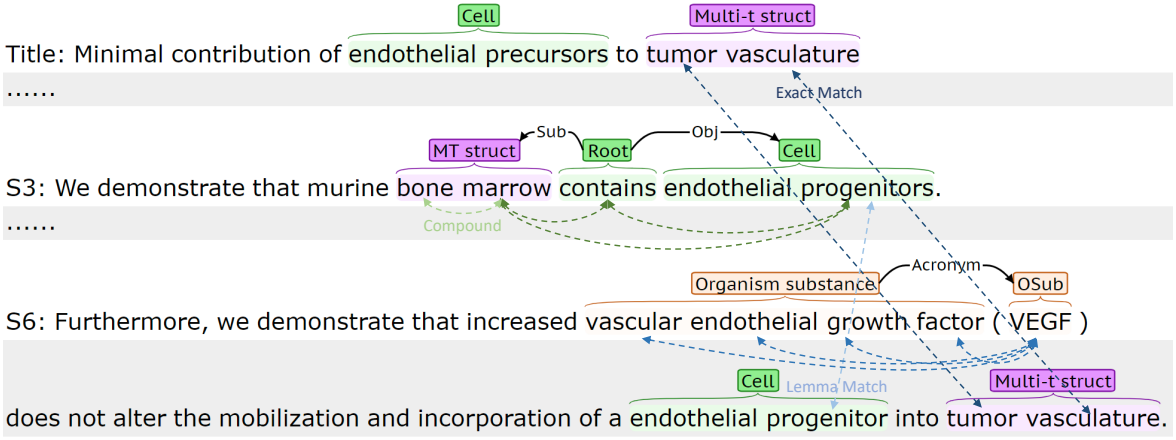

Figure 1: An example of NER with both discourse-level and sentence-level entity relations.

domain-specific NER, we plot learning curves and show that leveraging relations between entity mentions can effectively and consistently improve the NER performance when limited annotations are available.

2 Related Work

NER research has a long history and recent approaches (Yang and Zhang, 2018; Jiang et al., 2019; Jie and Lu, 2019; Li et al., 2020) using Neural Network models like BiLSTM-CNN-CRF (Ma and Hovy, 2016) and contextual embeddings such as BERT (Devlin et al., 2019) and FLAIR (Akbik et al., 2018) have improved the NER performance in the general domain to the human-level. However, the NER performance for specific domains is still moderate due to the challenges of limited annotations and dealing with complicated domain-specific contexts.

We aim to further improve NER performance by considering coreference relations and semantic relations between entity mentions. This is in contrast to the usual way of thinking about NER as an up-stream task conducted before coreference resolution or entity relation extraction. The idea aligns with recent works that conduct joint inferences among multiple information extraction tasks (Miwa and Bansal, 2016; Li et al., 2017; Bekoulis et al., 2018; Luan et al., 2019; Sui et al., 2020; Yuan et al., 2020), including NER, coreference resolution and relation extraction, by mining dependencies among the extractions. However, joint inference approaches require annotations for all the target tasks and aim to improve performance for all the tasks as well, while our lightweight approach aims to improve the performance of the basic NER task requiring no additional annotations (usually unavailable for specific domains).

Our approach is also related to several recent neural approaches for NER that encourage label dependencies among entity mentions. The Pooled FLAIR model (Akbik et al., 2019) proposed a global pooling mechanism to learn word representations. Dai et al. (2019) used a coreference layer with a regularizer to harmonize word representations. Closely related to our work, Qian et al. (2019) used graph neural nets to capture repetitions of the same word as well, but in a denser graph that includes edges between adjacent words and is meant to completely overlay the lower encoding layers. Memory networks (Gui et al., 2020; Luo et al., 2020) were also used to store and refine predictions of a base model by considering repetitions or co-occurrences of words. In addition, dependency relations have been commonly used to connect entities for relation extraction (Zhang et al., 2018; Bunescu and Mooney, 2005), but we aim to better infer the type of an entity by associating it with other closely related entities in a sentence.

3 Model Architecture

Our system with Entity Relation Graphs (EnRelG) mainly contains 5 layers as in Figure 2: an embedding layer, an encoding layer, a GNNs layer, a fusion layer, and a decoding layer.

3.1 Embedding Layer

We choose the BERT-base LM as our embedding layer. For domain-specific datasets, we use BioBERT (Lee et al., 2020) for the biomedical domain and SciBERT (Beltagy et al., 2019) for the planetary science domain. Specifically, for an input

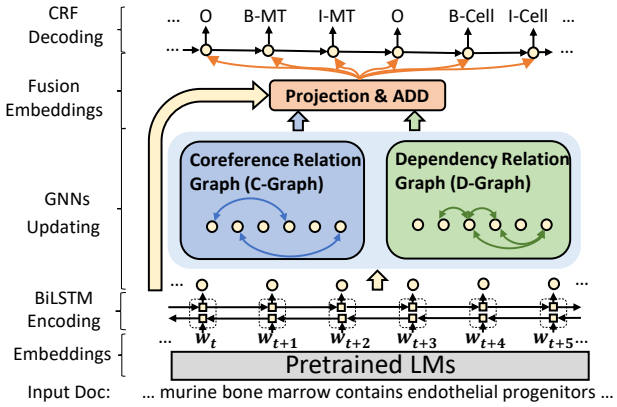

Figure 2: Overall Architecture of the EnRel-G system

document $D = [w_1, w_2, ..., w_n]$ with n words, the BERT model will output a contextual word embeddings matrix $\boldsymbol{E} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_n] \in \mathbb{R}^{n \times d_1}$ with a d_1 dimension vector for each word.

3.2 Encoding Layer

To capture the sequential context information, we use a BiLSTM layer to encode the word embeddings from the BERT model. We concatenate the forward and backward LSTM hidden states as the encoded representations and then obtain embedding matrix $\boldsymbol{E}^{lstm} = BiLSTM(\boldsymbol{E}) \in \mathbb{R}^{n \times d_2}$ with a d_2 dimension vector for each word.

3.3 Graph Neural Networks Layer

For the GNNs layer, we first introduce how to build Entity Relation Graphs using global coreference relations (coreference graph, C-graph) and local dependency relations (dependency graph, D-graph) between entities, and then describe how the GNNs model incorporates them into the word representations.

Coreference Relation Graph For each document, we build a graph $G^C = (\mathcal{V}, \mathcal{A}^C)$ based on coreference relations, in which $\mathcal{V}$ is a set of nodes denoting all the words in a document and $\mathcal{A}^C$ is the adjacency matrix. Specifically, we approximate the entity coreference relations using 3 syntactic coreference clues as in Figure 1: (1) *Exact Match*, two nouns are connected if they are the same, e.g., "tumor vasculature" in both the *Title* and *S6*; (2) *Lemma Match*, two nouns are linked together if they have the same lemma, e.g., "progenitors" and "progenitor" in the *S3* and *S6*; (3) *Acronym Match*, the acronym word is connected to all full expression words, e.g., "VEGF" and "vascular endothelial growth factor" in the *S6*. For each connected node pair (i, j), we set $\mathcal{A}^C_{i,j} = 1$. We also add a self-connection to each node ($\mathcal{A}^C_{i,i} = 1$) to maintain the words' original semantic information.

Dependency Relation Graph We build a Dependency Relation Graphs $G^D = (\mathcal{V}, \mathcal{A}^D)$ for each document based on sentence-level dependency relations. We first parse each sentence using the scispaCy[2] tool and then connect the following word pairs in the dependency tree: (1) *subject head word & object head word & their predicate*, we connect them to enhance the interactions between the entities from the subject and object. e.g., "marrow" and "progenitors" with the predicate "contains" in the *S3*; (2) *compound & head word*, we connect the compounds with their head words because they often both exist in an entity. e.g., the "bone" and "marrow" in the *S3*. Same as before, We set $\mathcal{A}^D_{i,j} = 1$ for each connect pair (i, j), and also add self-connection ($\mathcal{A}^D_{i,i} = 1$) for each node.

Then we update the encoded word embeddings with the entity relations graphs based on GNNs, particularly the GATs. Since nodes represent the words in a document, we initialize the node representations in the graphs from the encoding layer as $\boldsymbol{E}^{lstm} = [\boldsymbol{w}_1^{lstm}, \boldsymbol{w}_2^{lstm}, ..., \boldsymbol{w}_n^{lstm}]$. The graph attention mechanism updates the initial representation of node $\boldsymbol{w}_i^{lstm}$ to $\boldsymbol{w}_i^{gnn}$ by aggregating its neighbors' representations with their corresponding normalized attention scores.

$$\boldsymbol{w}_i^{gnn} = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k W^k \boldsymbol{w}_j^{lstm}\right) \tag{1}$$

As in equation (1), and we have K attention heads and concatenate ($\|$) them as the final representation. For head k, we weighted all the adjacent nodes ($\mathcal{N}_i$, obtained from the adjacent matrix $\mathcal{A}$) by W^k and and then aggregate them with the attention score α_{ij}^k. σ is the activation function LeakyReLU. The attention score α_{ij}^k is obtained as followed ($\boldsymbol{a}^T$ is a weight vector):

$$\alpha_{ij}^k = \frac{\exp\left(\sigma\left(\boldsymbol{a}^T\left(W^k \boldsymbol{w}_i^{lstm} \| W^k \boldsymbol{w}_j^{lstm}\right)\right)\right)}{\sum_{z \in \mathcal{N}_i} \exp\left(\sigma\left(\boldsymbol{a}^T\left(W^k \boldsymbol{w}_i^{lstm} \| W^k \boldsymbol{w}_z^{lstm}\right)\right)\right)} \tag{2}$$

For each of the two relation graphs, we use an independent graph attention layer. The output word representations from the two GATs are denoted as: $\boldsymbol{G}^C = [\boldsymbol{w}_1^{gnn(C)}, \boldsymbol{w}_2^{gnn(C)}, ..., \boldsymbol{w}_n^{gnn(C)}] \in \mathbb{R}^{n \times d_3}$ and $\boldsymbol{G}^D = [\boldsymbol{w}_1^{gnn(D)}, \boldsymbol{w}_2^{gnn(D)}, ..., \boldsymbol{w}_n^{gnn(D)}] \in \mathbb{R}^{n \times d_3}$, with d_3 dimension for each word.

[2] https://allenai.github.io/scispacy/

Methods	Datasets	
	AnatEM	Mars
Wagstaff et al. (2018)	–	94.5 / 77.7 / 85.3
NCRF++	83.40±0.34 / 76.96±0.46 / 80.05±0.12	91.28±1.08 / 80.57±0.55 / 85.59±0.23
FLAIR	81.07±0.29 / 75.28±0.57 / 78.06±0.39	90.67±1.02 / 81.45±1.41 / 85.81±0.62
Pooled FLAIR	82.11±0.50 / 77.55±0.40 / 79.76±0.34	87.79±1.31 / 86.57±1.10 / 87.17±0.17
Tuning Bio/SciBERT	83.94±0.40 / 83.12±0.30 / 83.53±0.32	90.93±0.66 / 88.99±1.61 / 89.95±0.64
EnRel-G (C)	84.65±0.67 / 83.69±0.31 / 84.17±0.41	91.21±1.05 / **89.35**±1.76 / 90.27±0.45
EnRel-G (D)	**84.98**±0.83 / 83.50±0.45 / 84.23±0.54	**92.66**±1.16 / 88.03±1.46 / 90.29±0.53
EnRel-G (CD)	84.86±0.50 / **83.96**±0.32 / **84.41**±0.24	92.57±1.00 / 88.65±1.50 / **90.57**±0.47

Table 1: Test results of baselines and our system (Average Precision/Recall/F1 Scores±standard deviation,%)[3]

3.4 Fusion Layer

Similar to Sui et al. (2019), we also use a fusion layer to blend the encoded word embeddings and the GNNs updated word embeddings. We first project these embeddings into the same hidden space using liner transformation and then add them, as in $\boldsymbol{F} = W_N \boldsymbol{E}^{lstm} + W_C \boldsymbol{G}^C + W_D \boldsymbol{G}^D$, where W_N, W_C, W_D are trainable weights. Then we will have a feature matrix $\mathbf{F} \in \mathbb{R}^{n \times d_4}$ for the n words blended with both the sequential context information and global entity relations.

3.5 Decoding Layer

Finally, a Conditional Random Field (CRF) (Lafferty et al., 2001) layer is used to decode the enriched embeddings $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_n]$ into a sequence of labels $y = \{y_1, y_2, ..., y_n\}$. In the training phrase, we optimize the whole model by minimizing the negative log-likelihood loss with respect to gold labels.

4 Experiments[4]

We test our model on two domain-specific datasets: the AnatEM (Pyysalo and Ananiadou, 2014) from the biomedical domain and the Mars (Wagstaff et al., 2018) from the planetary science domain. The AnatEM has annotated 12 types of entities in 1,212 documents with 13,701 entity mentions; the Mars has 117 longer documents with 4,458 entity mentions containing 3 types.

4.1 Baselines

NCRF++ (Yang and Zhang, 2018) is an open-source Neural Sequence Labelling Toolkit. We use the BiLSTM-CNN-CRF structrue as a baseline.

FLAIR (Akbik et al., 2018) is a character-level pretrained LM based on BiLSTM, which has been used in many NER systems (Jiang et al., 2019; Wang et al., 2019). We use the embeddings from it with a BiLSTM-CRF architecture as a baseline.

Pooled FLAIR (Akbik et al., 2019) is an extended version of the FLAIR model with global memory and pooling mechanism for the same word, which helps consistent predictions of coreferential entity mentions. We also use the embeddings from it with a BiLSTM-CRF architecture as a baseline.

Tuning Bio/SciBERT We also use Bio/SciBERT with a BiLSTM-CRF architecture as baselines for the AnatEM/Mars datasets, which do not have the GNNs layer or Fusion layer as compared with our system.

4.2 Results

To alleviate random turbulence, we train all the systems five times using different random seeds and evaluate their average performance on the test sets using the same script[5], as in the Table 1.

We can see that our system with both the global entity coreference and local dependency relations performs the best among all the systems. It improves the average F1 score by 0.88 points (84.41% vs. 83.53%) compared to BioBERT on the AnatEM, and 0.62 points (90.57% vs. 89.95%) compared to SciBERT on the Mars. Further, both the coreference and dependency relations help to improve the NER performance. Specifically, our model with either the coreference or dependency relation graph improves the F1 scores by 0.64 point or 0.7 point on the AnatEM dataset, and by 0.32 point or 0.34 point on the Mars dataset.

[3]Previous systems on the AnatEM dataset either evaluate the NER performance by head match or only evaluate the performance on span identification; therefore, so we do not include their results here.

[4]More details about the datasets, data preprocessing, and model settings can be found in the appendices.

[5]https://github.com/sighsmile/conlleval

Methods	Datasets	
	AnatEM	Mars
Tuning Bio/SciBERT	83.94±0.40 / 83.12±0.30 / 83.53±0.32	90.93±0.66 / 88.99±1.61 / 89.95±0.64
EnRel-G (D) (Key Edges Only)	83.79±0.70 / 83.39±0.39 / 83.59±0.40	91.71±0.63 / 88.30±0.86 / 89.97±0.33
EnRel-G (D) (Compound + Key Edges)	**84.98**±0.83 / 83.50±0.45 / **84.23**±0.54	**92.66**±1.16 / 88.03±1.46 / **90.29**±0.53
EnRel-G (D) (All Modifiers + Key Edges)	84.38±0.72 / **83.83**±0.31 / 84.10±0.40	91.06±1.94 / **89.19**±1.07 / 90.11±0.55
EnRel-G (D) (All Dependency Edges)	84.32±0.36 / 83.52±0.44 / 83.92±0.30	90.71±2.85 / 89.62±1.87 / 90.16±1.23

Table 2: Edge Selection in the Dependency Graph (Average Precision/Recall/F1 Scores±standard deviation,%)

4.3 Learning Curves

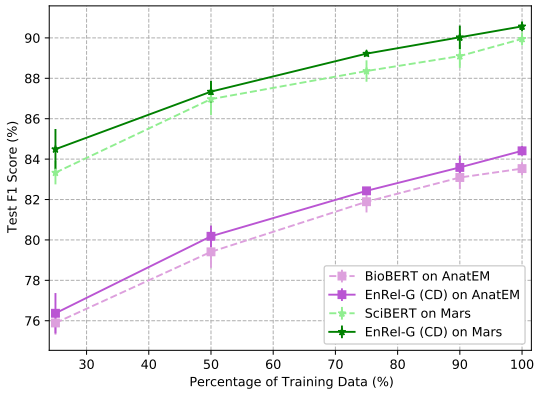

Figure 3: Learning Curves, each point shows the average performance of 5 system runs.

One main limitation of domain-specific NER systems is the lack of annotations, therefore, it is vital to make the best use of labeled data. The learning curves (Figure 3) shows that leveraging the relations between entity mentions can effectively elevate the NER performance to a higher level even when only a tiny amount of labeled data (a quarter of training data) is available, and this is true on both the AnatEM dataset and the Mars dataset.

4.4 Analysis of Computation Cost

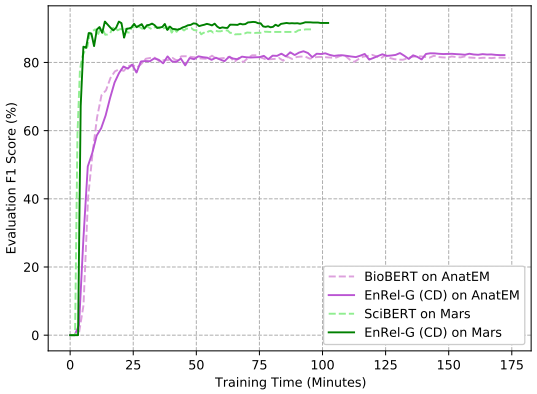

Figure 4: Comparison of Training Time

Although fine-tuning pretrained LMs has improved the performance of many NLP tasks, one limitation is the increase of training time. Therefore, it is important to build computing efficient models based on pretrained LMs. As shown in Figure 4, our model with the GNNs layer does not increase the time cost for fine-tuning the BERT models. The training time of methods with or without the GNNs layer is similar.

4.5 Edge selection in the Dependency Graph

To build the sentence-level dependency graph, we selected only two types of dependency relations: between the subject, object and their predicate (*Key Edges*) and between a compound modifier and its head word. As shown in the Table 2, we also tried to connect all the modifiers with their head word and found that this yields slightly worse performance, and the reason may be that many modifiers other than compounds are not entities themselves. In addition, including all the dependency edges also yields worse performance than using the two selected types of dependency relations, probably for the same reason that many of the nodes in a dependency tree are not parts of entity mentions and many dependency relations do not directly contribute to capturing relations between entities.

5 Conclusion

In this work, we explicitly capture the global coreference and local dependency relations between entity mentions, and use graph neural nets to incorporate the relations to improve domain-specific NER tasks. Experimental results on two datasets show the effectiveness of this lightweight approach. We also find that the selection of entity relations is important to the system performance. Future work may consider about using GNNs to incorporate external knowledge for performance improvement.

Acknowledgments

This work was supported by a gift from Bosch Research and NSF Award IIS-1909255.

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.

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.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34–45.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731.

Zeyu Dai, Hongliang Fei, and Ping Li. 2019. Coreference aware representation learning for neural named entity recognition. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4946–4953. International Joint Conferences on Artificial Intelligence Organization.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tao Gui, Jiacheng Ye, Qi Zhang, Yaqian Zhou, Yeyun Gong, and Xuanjing Huang. 2020. Leveraging document-level label consistency for named entity recognition. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3976–3982. International Joint Conferences on Artificial Intelligence Organization. Main track.

Yufan Jiang, Chi Hu, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2019. Improved differentiable architecture search for language modeling and 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 3585–3590, Hong Kong, China. Association for Computational Linguistics.

Zhanming Jie and Wei Lu. 2019. Dependency-guided LSTM-CRF for 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 3862–3872, Hong Kong, China. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. 2017. A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, 18(1):1–11.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. 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.

Hu Linmei, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. 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 4823–4832.

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.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical contextualized representation for named entity recognition. In *AAAI*, pages 8441–8448.

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.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Tomoko Ohta, Sampo Pyysalo, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Open-domain anatomical entity mention detection. In *Proceedings of the workshop on detecting structure in scholarly discourse*, pages 27–36.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Sampo Pyysalo and Sophia Ananiadou. 2014. Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875.

Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, Han-Cheol Cho, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18):i575–i581.

Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. 2019. GraphIE: A graph-based framework for information extraction. 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 751–761, Minneapolis, Minnesota. Association for Computational Linguistics.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage lexical knowledge for chinese named entity recognition via collaborative graph network. 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 3821–3831.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Xiangrong Zeng, and Shengping Liu. 2020. Joint entity and relation extraction with set prediction networks. *arXiv preprint arXiv:2011.01675*.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Kiri Wagstaff, Raymond Francis, Thamme Gowda, You Lu, Ellen Riloff, Karanjeet Singh, and Nina Lanza. 2018. Mars target encyclopedia: Rock and soil composition extracted from the literature. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training named entity tagger from imperfect annotations. 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 5154–5163, Hong Kong, China. Association for Computational Linguistics.

Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. 2020. A relation-specific attention network for joint entity and relation extraction. In *International Joint Conference on Artificial Intelligence 2020*, pages 4054–4060. Association for the Advancement of Artificial Intelligence (AAAI).

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Appendices

Appendix A: Dataset Details

The AnatEM (Pyysalo and Ananiadou, 2014) dataset is an extended Anatomical Entity Mention corpus combining both the Anatomical Entity Mention (AnEM) (Ohta et al., 2012) dataset and Multi-level Event Extraction (MLEE) (Pyysalo et al., 2012) corpus. All the documents are selected from PubMed[6] abstracts or full-text papers. AnatEM is manually annotated by biological experts and it has 12 types of entities annotated, namely *Anatomical System, Cancer, Cell, Cellular Component, Developing Anatomical Structure, Immaterial Anatomical Entity, Multi-tissue Structure, Organ, Organism Subdivision, Organism Substance, Pathological Formation, Tissue*. In total, this dataset consists of 1,212 documents and 13,701 entities annotated.

[6] https://pubmed.ncbi.nlm.nih.gov/

Datasets		#Doc	#Words	#Entities	#Words/Doc
AnatEM	Train	606	153,823	6,946	254
	Dev	202	58,785	2,139	291
	Test	404	99,976	4,616	247
	Total	1,212	312,584	13,701	258
Mars	Train	62	99,952	2,431	1,612
	Dev	20	33,743	906	1,687
	Test	35	58,392	1,121	1,668
	Total	117	192,087	4,458	1,642

Table 3: Statistics of the AnatEM and Mars datasets.[7]

Mars is from the scientific literature domain, and it is about planetary science. All documents come from the Lunar and Planetary Science Conference (LPSC)[8], and the entity mentions are annotated manually. It has 3 types of entities: *Element*, *Mineral*, *Target*. The corpus consists of 117 documents. 62 of them are from LPSC 2015 and they are for training and 55 of them are from LPSC 2016 for evaluation. Same as previous work, we divide the 2016 documents into a validation set with 20 documents and a testing set with 35 documents.

Appendix B: Data Preprocessing

We want our model to take advantage of the document-level information, but some of the documents are extremely too long. Moreover, the BERT model also has a limitation of 512 subtokens for input texts. So we need to split the long documents. Besides, the BERT language model needs a big enough batch size (e.g., 16 or 32) to be well fine-tuned, which is also a burden for the GPU memory consumption. In consideration of these restrictions, we limit the max subtoken count of a split document to 128 in the data preprocessing. Future work with more computing resources may try longer input documents.

Moreover, we also add the POS and Dependency Tree information into the data using scispaCy for constructing the Coreference Graph and the Dependency Graph in our model.

Appendix C: Model Settings

For the **NCRF++** baseline, we use one layer of BiLSTM for word sequence representation with 300-dim Glove (Pennington et al., 2014) embeddings, four layers of CNN for character sequence representation with 50-dim random initialized character embeddings, and a CRF layer for inference.

Methods	Optimizer	Learning Rate	Batch Size
NCRF++	SGD	1e-2	10
(pooled) FLAIR	Adam	2e-3	8
Tuning Bio/SciBERT	Adam	5e-5	32
EnRel-G	Adam	5e-5	32

Table 4: Model Settings

For the **FLAIR** and **Pooled FLAIR** baselines, we use the PubMed version (pretrained on the biomedical corpus) for the AnatEM dataset and the general English version (pretrained on the English news articles) for the Mars dataset. Particularly, for the Pooled FLAIR model, we set the *mean* pooling mechanism to calculate the average of embeddings for multiple occurrences of a word, and then use it as the representation for the word.

For the **Tuning BERT** baselines, we use *BioBERT-Base v1.1* for the AnatEM dataset and *SciBERT-scivocab-uncased* for the Mars dataset.

For our **EnRel-G** system, we keep the embeddings layer the same as the Tuning BERT baselines. As for the GNNs layer, we use one layer of the graph attention mechanism with 4 heads, and each head has a hidden dimension of 128.

For the optimization related parameters, as in the Table 4, we mainly use the recommended settings for the baseline models. For our EnRel-G system, we keep the same parameters as in the Tuning BERT baseline for fair comparison.

We train all the systems on a single Nvidia GEFORCE GTX 2080Ti GPU. We set the maximum epoch as 100 and use the best-performed model on the development set to evaluate the test data.

[7] We remove the redundantly annotated entities in the Mars.

[8] https://www.hou.usra.edu/meetings/