Relation Discovery via Graph Neural Networks in the Era of Large Language Model

Zhiqiang Wang[1,2], Yiping Yang[1], and Junjie Ma[3]

[1] Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] China Academy of Electronics and Information Technology, Beijing, China
{wangzhiqiang2019,yiping.yang}@ia.ac.cn, junjiema@tsinghua.edu.cn

Abstract. The purpose of relation extraction is to discern semantic relationships between entities within sentences, positioning it as a classification task. In recent years, the integration of graph neural networks (GNNs) with large language models (LLMs) has shown promise in addressing relation extraction challenges. However, prevalent graph-centric approaches still grapple with notable challenges: firstly, they often fail to harness dependency relation information comprehensively; secondly, a standardized protocol for pruning dependency trees in the context of these models remains elusive. To mitigate these concerns, we propose a two-phase graph convolutional network (GCN) tailored for relation extraction, leveraging the contextual understanding provided by a large language model. In the initial phase of our model, we integrate node representations derived from the LLM, dependency relation type representations, and dependency type weights to collectively derive novel node embeddings, thereby exhaustively leveraging dependency relation information. Transitioning to the second phase, we leverage the adjacency matrix extracted from the dependency tree to execute graph convolution operations, guided by insights from the LLM. This dual-phase approach enables our model to dynamically and autonomously prune the dependency tree while benefiting from the contextual richness of the large language model. We evaluated our methodology on two publicly accessible datasets, and the outcomes demonstrate that our model surpasses prior research in terms of the F1 score, attaining state-of-the-art performance. Furthermore, supplementary ablation experiments underscore the efficacy of each constituent within our proposed model, validating the contribution of both the GCN and the large language model.

Keywords: Relation discovery · Large language model · Graph neural networks.

1 Introduction

The goal of relation extraction (RE) is to identify and extract entities from text, along with their semantic relationships, forming relation triples in the format

of (subject, relation, object). This task holds significant importance in various downstream applications within natural language processing (NLP), including information extraction [8], knowledge graph construction [18], user profiling [23], sentiment analysis [4], automatic question answering [14], and more. Consequently, it has garnered considerable attention in recent years.

In recent times, the field of relation triple extraction has witnessed several research endeavors. Historically, the evolution of relation extraction can be broadly categorized into three stages. Initially, feature-based methods[8] dominated, combining numerous lexical, grammatical, and semantic features with machine learning algorithms. However, the selection of features played a pivotal role, and different combinations significantly influenced model performance. Subsequently, kernel-based methods[27] emerged, specifying similarity measures between samples. Yet, the design of the kernel function remained a critical factor affecting performance.

More recently, neural network-based approaches have achieved remarkable success in relation extraction tasks[12], drawing increasing attention from researchers. This shift from traditional to deep learning methods has been widespread. Among these neural methods, various architectures have played crucial roles, with graph convolutional networks being particularly prominent. For instance, Xu et al.[22] introduced the shortest dependency path method, integrating segmented LSTM models for relation extraction. Guo et al.[6] directly applied graph convolutional neural networks, while Tian et al.[16] incorporated edge information using a multi-head attention mechanism to refine adjacency matrices.

Despite their advancements, these methods share common limitations: they often overlook the specific types of dependency relationships when constructing adjacency matrices, fail to consider the varying impacts of different dependency types, and lack a unified standard for pruning operations.

To tackle these challenges, we propose an innovative approach leveraging the capabilities of large language models (LLMs) and graph-based representations for relation extraction. Our method begins by performing dependency analysis on input sentences to obtain dependency trees. Instead of solely relying on adjacency matrices, we also construct a dependency relation type matrix ($\mathbf{D}$) and a dependency type weight matrix ($\mathbf{W}$). The dependency relation type matrix labels the types of relationships between words, while the dependency type weight matrix captures the varying impacts of these relationships on sentence semantics.

In our approach, we integrate these matrices with node representations within a graph neural network framework. This integration allows for more comprehensive information exchange between nodes, enhancing the model's ability to capture nuanced semantic relationships. By leveraging the power of LLMs for initial text representation and graph-based methods for relational reasoning, our model achieves robust performance in relation extraction tasks.

Experiment on the DDIExtraction 2013, ChemProt and GAD datasets shows that our proposed method is effective and outperforms the performance of the

best existing models. Further ablation experiments also demonstrate the effectiveness of our proposed model.

The contributions of this paper are summarized as follows:

- We incorporate dependency relation types and dependency type weights, leveraging the rich syntactic information provided by dependency analysis.
- We present a novel integration of large language models and graph-based methods for relation extraction, enabling enhanced semantic understanding.
- Through extensive experiments on public datasets, we achieve state-of-the-art results, demonstrating the superiority of our proposed method.

The remainder of this paper is organized as follows. Section2 reviews the relevant background and literature. Section3 provides a detailed description of our proposed method. Experimental settings and results are presented in Section**??**. Finally, Section**??** concludes the paper with a summary and future directions.

2 Related Work

AGGCN is an early effort that introduces graph convolutional networks for relation extraction, leveraging the capabilities of early language models [6]. However, it falls short by failing to distinguish between different types of dependencies, treating all dependencies equally, which subsequently hinders the model's performance. Tian et al. [16] improve upon this by paying attention to dependency types. In calculating the adjacency matrix $\mathbf{A}$, they replace the traditional $\mathbf{A}$ with a result deduced from the dependency type and two dependency words, and incorporate a generalized attention mechanism. They evaluate four pruning strategies in the generated dependency matrix: 'Local', 'Global', 'Full', and 'Local+Global'. For 'Global', they employ the shortest path strategy, searching for the shortest dependent path between two entities. Notably, solving for the shortest dependency path, regardless of using BFS or DFS algorithms, involves recursive strategies that can incur significant computational costs, particularly as sentence length increases, leading to decreased computational efficiency. Xu et al. [22] also adopt a combination of the shortest path and LSTM model, reasoning that finding the shortest path between two entities yields the sentence's skeleton, crucial for sentence analysis and entity relation extraction. They divide the sentence skeleton into two parts by the root word and apply an LSTM model to each part, marking the first use of the shortest path in this context. Yet, for long sentences, finding the shortest path between entities remains computationally expensive. Furthermore, entities are not always individual words. Zhang et al. [28] argue that aggressive pruning of the dependency tree can lead to the loss of negative meanings and incomplete information expression, undermining model robustness. They propose three pruning strategies to address this. Yu et al. [25] similarly caution that manual pruning may omit useful information. Another method [3] enhances this by extracting two types of dependency information for each word in an entity: "intra-entity" dependency and "cross-entity"

dependency, which represents the dependency path between two entities. This method also uses the shortest dependency path.

Our work builds upon these foundations but distinguishes itself by considering both the words in the sentence and the types of dependency relationships between them. To maximize the preservation of semantic information in the sentence, we refrain from pruning operations, aligning with advanced language model practices that prioritize comprehensive information retention.

3 Our Proposed Approach

In this section, we will first introduce the task definition, followed by a detailed description of the proposed model. The overall architecture of our model is illustrated in Fig. 1.

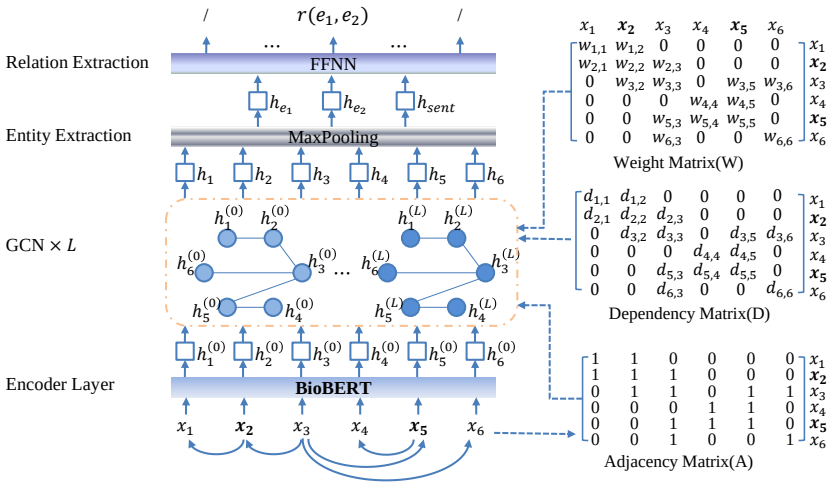

Fig. 1. Overview of our model architecture illustrated with an example sentence.

3.1 Task Definition

Given a pre-defined relation set $\mathcal{R} = \{r_1, r_2, \cdots, r_m\}$ contains m types, and a sentence $s = \{x_1, x_2, \cdots, x_n\}$ composed of n words. Note that the entities in the sentence have been labeled. The goal of the task is to identify the semantic relationship between the two entities and form a triple (subject, relation, object). An entity consists of one or more consecutive words in the sentence.

3.2 Large language models for Label

Large language models can have relational labels when processing text, which enables them to understand and analyze text content more deeply. Traditional classification based methods often fail to fully utilize label information and rely solely on feature engineering and classification algorithms when handling relationship extraction tasks. However, with the development of large-scale language models, we can use these models to more tightly integrate given sentences with label information.

Taking the sentence 'Steve Jobs founded Apple Inc.' as an example, if we know there is a label 'founded', we can use large language models for more in-depth analysis. We can ask the big model the following question: "What is the relationship between Steve Jobs and Apple Inc.? Is it a founding relationship? Why?" The large language model will provide an explanation containing rich semantic information based on its powerful language understanding and generation capabilities. This explanation not only confirms that the relationship between two entities is' established ', but may also contain other relevant information about this relationship, such as time, place, background, etc.

Next, we can combine the original sentence "Steve Jobs founded Apple Inc." with the explanatory sentence provided by the large language model to form a new and more complete sentence. This new sentence not only contains the original information, but also incorporates rich semantic explanations from large language models. Through this approach, we can more accurately discover relationships and extract complex semantic relationships hidden in the text. This method not only improves the accuracy of relationship extraction, but also enhances the interpretability of the model, allowing us to better understand how the model performs relationship extraction.

3.3 Encoder

To obtain the context-aware representation of each word in the input sentence $s = \{x_1, x_2, \cdots, x_n\}$. We choose BERT [5] as our sentence encoder. This process can be expressed using the following formula:

$$\{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n\} = \text{BERT}(\{x_1, x_2, \cdots, x_n\} \tag{1}$$

where $\mathbf{h}_i \in \mathbb{R}^d$ is the contextual representation of the i^{th} word in sentenc s and d is the embedding size.

3.4 Graph Neural Network

Standard Graph Convolutional Networks The standard graph convolutional networks can be expressed by the following formula:

$$\mathbf{h}_i^l = \sigma\left(\sum_{j=1}^{n} a_{i,j}\left(\mathbf{W}^l \mathbf{h}_j^{l-1} + \mathbf{b}^l\right)\right) \tag{2}$$

where $\mathbf{h}_i^l$ and $\mathbf{h}_i^{l-1}$ denote the contextual representations of the i^{th} word in the l^{th} and $(l-1)^{th}$ layers, respectively, $\mathbf{W}^l$ and $\mathbf{b}^l$ are trainable matrix and bias, $a_{i,j}$ denote the entry at the i^{th} row and j^{th} column of the adjacency matrix $\mathbf{A}$ and σ denotes activation function, such as ReLU. Graph convolutional networks can be seen as nodes in the l^{th} layer gather information from their neighboring nodes in the $(l-1)^{th}$ layer to represent themselves.

Adjacency Matrix Given a sentence $s = \{x_1, x_2, \cdots, x_n\}$ with n words, we first use the existing natural language processing toolkit CoreNLP[4] to perform dependency parsing, which identifies the dependency relationships between words, as shown in Fig. 1(b). Based on the generated dependency tree, we can form an adjacency matrix $\mathbf{A} = (a_{i,j})_{n \times n}$, where the entry $a_{i,j} = a_{j,i} = 1$ if there is a relationship between words x_i and x_j; otherwise, $a_{i,j} = a_{j,i} = 0$. Note that the matrix $\mathbf{A}$ is symmetric.

Dependency Relation Type Matrix Because the adjacency matrix $\mathbf{A}$ can only reflect whether there is an dependency relationship between two words but cannot capture other information such as dependency relation type, so we construct a corresponding matrix $\mathbf{D}$ for representing dependency relation types. In this matrix, each entry is an dependency relation type, such as 'nsubj', 'amod' and so on. In order to enable interaction between the dependency relation information and the node representations, we embed each dependency relation type to $\mathbb{R}^d$, the same dimens with word embedding.

Dependency Relation Weight Matrix To explicitly capture the varying importance of each dependency relation type in representing information within a sentence, we construct a matrix $\mathbf{W}$ that corresponds to $\mathbf{D}$, called dependency relation weight matrix. It serves as a weighting mechanism for the dependency relation types, controlling the interaction between the dependency relation information and the node representations. Similar to $\mathbf{D}$, we also embed each of its elements to d-dimensional space.

Calculation of Graph Convolution The graph convolution operation in the first step primarily focuses focuses on the interaction between node information and dependency relation information, as shown in Fig. 1(c). Since each word in the sentence can have dependency relationships with other words, the representation of node i can be extended to an $n \times d$ matrix. Specifically, if there exists a dependency relation d between node i and node j, the j^{th} row of node i can be computed using the following formula:

$$\tilde{\mathbf{h}}_{i,j}^{l-1} = \mathbf{h}_i^{l-1} + \mathbf{w}_{i \to j} \odot \mathbf{d}_{i \to j} \tag{3}$$

[4] https://stanfordnlp.github.io/CoreNLP/

Similarly, for node j, its i^{th} row can be computed using the following formula:

$$\tilde{\mathbf{h}}_{j,i}^{l-1} = \mathbf{h}_{j}^{l-1} + \mathbf{w}_{j\to i} \odot \mathbf{d}_{j\to i} \tag{4}$$

where $\mathbf{h}_{i}^{l-1}$, $\mathbf{h}_{j}^{l-1}$ denote the i^{th} and j^{th} node representation in the $(l-1)^{th}$ layer, respectively, $\mathbf{w}_{i\to j}$, $\mathbf{w}_{j\to i}$, $\mathbf{d}_{i\to j}$ and $\mathbf{d}_{j\to i}$ denote the embedding of relation d and weight w, respectively, $\odot$ denotes the element-wise multiplication.

Through this operation, the model fully leverages both node information and dependency relation information.

The second step is primarily dedicated to performing convolutional operation, as illustrated in Fig. 1(c). This process can be expressed using the following formula:

$$\mathbf{h}_{i}^{(l)} = \sigma\left(\sum_{j=1}^{n} a_{i,j}\left(\mathbf{W}^{(l)}\tilde{\mathbf{h}}_{j}^{(l-1)} + \mathbf{b}^{(l)}\right)\right) \tag{5}$$

the relevant parameters are similar to Eq. 2.

4 Experiments

4.1 Dataset

We validate our proposed method on three public datasets, which are ChemProt, DDI and GAD. ChemProt [9] consists of PubMed abstracts corpora with five high-level chemicalprotein interaction annotations. DDI [7] studies drug-drug interaction and specializes in pharmacovigilance built from PubMed abstracts. GAD [2] is a semi-labeled dataset created using Genetic Association Archive and consists of gene-disease associations. The statistics of the dataset are shown in Table 1.

Table 1. Dataset statistics

Dataset	Relation	Train	Dev	Test	#relation
ChemProt	chemical-gene	18305	11268	15745	5
DDI	drug-drug	25296	2496	5716	4
GAD	disease-gene	4261	535	534	2

4.2 Evaluation Criteria

To measure the performance of the model, like existing works, we use precision, recall, and F1-score(%)-score as evaluation metrics. Our model is implemented with PyTorch and we adopt AdamW [11] as our optimizer. All experiments are conducted on a single RTX 3090 GPU.

4.3 Experimental Results

Our experimental results are shown in Table 2 and Table 3. We also compare the existing results with our experimental results. From Table 2, it can be seen that, for ChemProt, we exceed the current best method BioM-BERT [1] by 1%, and for DDI, our method exceeds the current best method NBR [21] by 1%. It should be noted that the NBR method is based on large language model (LLM), and compared to our method, our method not only achieves competitive results, but also does not incur any additional costs of calling the large language model such as ChatGPT. At the same time, we can also see that our method is significantly superior to traditional annotation frameworks, such as NovelTagging [29], CasRel [20], TPLinker [19] and UIM [15].

For GAD, since previous works only provide F1-score(%)-score, for comparison purposes, we only report the experimental results of our F1-score(%)-score. From the Table 3, it can be seen that compared with the previous optimal method, our method also exceeded 0.75%. This fully demonstrates the effectiveness of our model.

Table 2. Comparison with other methods on ChemProt and DDI.

Model	**ChemProt**			**DDI**		
	Pre.(%)	**Rec.(%)**	**F1-score(%)**	**Pre.(%)**	**Rec.(%)**	**F1-score(%)**
NovelTagging [29]	63.7	47.8	54.6	76.3	72.2	74.2
CasRel [20]	52.7	55.6	54.1	66.9	64.8	65.8
TPLinker [19]	61.6	59.5	65.0	71.7	74.0	72.8
UIM [15]	69.2	68.5	68.8	75.2	82.0	77.7
BioM-BERT [1]	-	-	80	-	-	82.0
NBR [21]	-	-	79.0	-	-	84.0
Ours	79.0	83.0	**81.0**	84.0	86.0	**85.0**

Table 3. Comparison with other methods on GAD.

Model	**F1-score(%)**
Bio-BERT [10]	79.83
PubMed-BERT [17]	82.34
Sci-Five [13]	79.21
KeBioLM [26]	84.3
BioLink-BERT [24]	84.39
NBR [21]	83.75
Ours	**85.14**

4.4 Ablation Study

To verify the effectiveness of each module in the proposed model, we conduct ablation studies on DDI. From the experimental results, it can be seen that increasing weights and pruning operations have a significant impact on the performance of the model. In the absence of weights and pruning, the performance will decrease by more than 5% from the F1-score(%)-score. We can also see that weight operation has a more significant impact on performance than pruning operation, because under the condition of only weight, performance decreases by 2.85%, while under the condition of only pruning, performance decreases by 3.06%.

Table 4. Results of ablation experiments.

weight	pruning	**Pre.(%)**	**Rec.(%)**	**F1-score(%)**
w	w	84.3	85.8	85.04
w	w/o	80.1	84.4	82.19
w/o	w	81.28	82.7	81.98
w/o	w/o	79.5	80.3	79.9

4.5 Further Research

In model [21], the author cleverly employed the mask technique, which replaces key entity words in sentences with unified, non-specific markers. For example, in the ChemProt dataset, the chemical and gene entities in the sentence are replaced with CHEMICAL and GENE. This operation effectively eliminates the direct impact of entity words on the model processing process, as it allows the model to focus more on sentence structure and contextual information rather than being distracted by specific entity names, which may improve the accuracy and generalization ability of the model in identifying the relationship between chemicals and genes.

In our experiment, we also conducted relevant research and obtained specific experimental results as shown in Table 5.

Table 5. The F1-score(%) based on the experimental results using entity replacement.

	DDI	**ChemProt**	**GAD**
base	81.0	85.0	85.14
Rep.	79.9	83.2	82.58

From the data in Table 5, it can be seen that using the mask method (i.e. replacing entity words in sentences with specific markers) did not bring the

expected performance improvement in our experiment, but instead led to a significant decrease in F1 score. On three datasets x, y, and z, there was a decrease of 1.1%, 1.8%, and 2.56%, respectively. We speculate that the possible reasons for this phenomenon are: 1) loss of semantic information: replacing specific entity words with generic markers (such as CHEMICAL and GENE) may result in a significant loss of semantic information. These entity words often contain rich contextual information and domain specific knowledge, which are crucial for models to understand and judge relationships in sentences. When this information is stripped away, the model may find it difficult to accurately capture and infer complex relationships between entities. 2) Limited model generalization ability: Replacement strategies may limit the model's generalization ability. During the training process, the model establishes a relational model by learning specific entity words and their contexts. When unseen entity words appear in the test data, if the model overly relies on specific representations of entity words, it may not be able to adapt well to these new situations. The replacement strategy further exacerbates this problem as it removes the differences between entity words, making it more difficult for the model to learn effective generalization rules from limited training data.

5 Conclusion

In this paper, we introduce a graph convolutional network model that combines dependency and weight (WGCN) to extract relation triples. Unlike the previous approaches, our method performs this task in two steps: establishing the adjacency matrix according to the dependency between words and the corresponding dependency matrix and weight matrix. In order to increase the robustness of the model, we prune the input dependency tree. Experiments on benchmarks show that our proposed method is effective. Next, we will explore the application of the proposed method in many downstream tasks of natural language processing.

References

1. Alrowili, S., Vijay-Shanker, K.: Biom-transformers: building large biomedical language models with bert, albert and electra. In: Proceedings of the 20th workshop on biomedical language processing. pp. 221–227 (2021)
2. Bravo, À., Piñero, J., Queralt-Rosinach, N., Rautschka, M., Furlong, L.I.: Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. BMC bioinformatics **16**, 1–17 (2015)
3. Chen, G., Tian, Y., Song, Y., Wan, X.: Relation extraction with type-aware map memories of word dependencies. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021. Findings of ACL, vol. ACL/IJCNLP 2021, pp. 2501–2512. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.findings-acl.221, https://doi.org/10.18653/v1/2021.findings-acl.221

4. Denecke, K.: Sentiment Analysis in the Medical Domain. Springer (2023). https://doi.org/10.1007/978-3-031-30187-2, https://doi.org/10.1007/978-3-031-30187-2
5. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423, https://doi.org/10.18653/v1/n19-1423
6. Guo, Z., Zhang, Y., Lu, W.: Attention guided graph convolutional networks for relation extraction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 241–251 (2019)
7. Herrero-Zazo, M., Segura-Bedmar, I., Martínez, P., Declerck, T.: The DDI corpus: An annotated corpus with pharmacological substances and drug-drug interactions. J. Biomed. Informatics **46**(5), 914–920 (2013)
8. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July 21-26, 2004 - Poster and Demonstration. ACL (2004), https://aclanthology.org/P04-3022/
9. Krallinger, M., Rabal, O., Akhondi, S.A., Pérez, M.P., Santamaría, J., Rodríguez, G.P., Tsatsaronis, G., Intxaurrondo, A., López, J.A., Nandal, U., et al.: Overview of the biocreative vi chemical-protein interaction track. In: Proceedings of the sixth BioCreative challenge evaluation workshop. vol. 1, pp. 141–146 (2017)
10. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
11. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
12. Pérez-Pérez, M., Ferreira, T., Igrejas, G., Fdez-Riverola, F.: A deep learning relation extraction approach to support a biomedical semi-automatic curation task: The case of the gluten bibliome. Expert Syst. Appl. **195**, 116616 (2022). https://doi.org/10.1016/j.eswa.2022.116616, https://doi.org/10.1016/j.eswa.2022.116616
13. Phan, L.N., Anibal, J.T., Tran, H., Chanana, S., Bahadroglu, E., Peltekian, A., Altan-Bonnet, G.: Scifive: a text-to-text transformer model for biomedical literature. arXiv preprint arXiv:2106.03598 (2021)
14. Sheng, S., Singh, A., Goswami, V., Magana, J., Thrush, T., Galuba, W., Parikh, D., Kiela, D.: Human-adversarial visual question answering. Advances in Neural Information Processing Systems **34**, 20346–20359 (2021)
15. Tan, H., Yang, Z., Ding, Z., Wang, Z., Luo, L., Wang, L., Zhang, Y., Liu, W., Lin, H., Wang, J.: Joint biomedical entity and relation extraction with unified interaction maps. In: 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). pp. 1437–1442. IEEE (2023)
16. Tian, Y., Chen, G., Song, Y., Wan, X.: Dependency-driven relation extraction with attentive graph convolutional networks. 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). pp. 4458–4471 (2021)

17. Tinn, R., Cheng, H., Gu, Y., Usuyama, N., Liu, X., Naumann, T., Gao, J., Poon, H.: Fine-tuning large neural language models for biomedical natural language processing. Patterns **4**(4) (2023)
18. Wang, X., Huang, T., Wang, D., Yuan, Y., Liu, Z., He, X., Chua, T.S.: Learning intents behind interactions with knowledge graph for recommendation. In: Proceedings of the Web Conference 2021. pp. 878–887 (2021)
19. Wang, Y., Yu, B., Zhang, Y., Liu, T., Zhu, H., Sun, L.: Tplinker: Single-stage joint extraction of entities and relations through token pair linking. arXiv preprint arXiv:2010.13415 (2020)
20. Wei, Z., Su, J., Wang, Y., Tian, Y., Chang, Y.: A novel cascade binary tagging framework for relational triple extraction. arXiv preprint arXiv:1909.03227 (2019)
21. Xu, J., Ma, M.D., Chen, M.: Can nli provide proper indirect supervision for low-resource biomedical relation extraction? arXiv preprint arXiv:2212.10784 (2022)
22. Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., Jin, Z.: Classifying relations via long short term memory networks along shortest dependency paths. In: Proceedings of the 2015 conference on empirical methods in natural language processing. pp. 1785–1794 (2015)
23. Yan, Q., Zhang, Y., Liu, Q., Wu, S., Wang, L.: Relation-aware heterogeneous graph for user profiling. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 3573–3577 (2021)
24. Yasunaga, M., Leskovec, J., Liang, P.: Linkbert: Pretraining language models with document links. arXiv preprint arXiv:2203.15827 (2022)
25. Yu, B., Mengge, X., Zhang, Z., Liu, T., Yubin, W., Wang, B.: Learning to prune dependency trees with rethinking for neural relation extraction. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 3842–3852 (2020)
26. Yuan, Z., Liu, Y., Tan, C., Huang, S., Huang, F.: Improving biomedical pretrained language models with knowledge. arXiv preprint arXiv:2104.10344 (2021)
27. Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. J. Mach. Learn. Res. **3**, 1083–1106 (2003), http://jmlr.org/papers/v3/zelenko03a.html
28. Zhang, Y., Qi, P., Manning, C.D.: Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2205–2215 (2018)
29. Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., Xu, B.: Joint extraction of entities and relations based on a novel tagging scheme. arXiv preprint arXiv:1706.05075 (2017)