Combinatory Grammar Tells Underlying Relevance among Entities

Yuanhe Tian[♠♥], **Yan Song**[♠†]
[♠]University of Science and Technology of China [♥]University of Washington
[♥]yhtian@uw.edu [♠]clksong@gmail.com

Abstract

Relation extraction (RE) is an important task in natural language processing which aims to annotate the relation between two given entities, which requires a deep understanding of the running text. To import model performance, existing approaches leverage syntactic information to facilitate the relation extraction process, where they mainly focus on dependencies among words while paying limited attention to other types of syntactic structure. Considering that combinatory categorial grammar (CCG) is a lexicalized grammatical formalism that carries the syntactic and semantic knowledge for text understanding, we propose an alternative solution for RE that takes advantage of CCG to detect the relation between entities. In doing so, we perform a multi-task learning process to learn from RE and auto-annotated CCG supertags, where an attention mechanism is performed over all input words to distinguish the important ones for RE with the attention weights guided by the supertag decoding process. We evaluate our model on two widely used English benchmark datasets (i.e., ACE2005EN and SemEval 2010 Task 8 datasets) for RE, where the effectiveness of our approach is demonstrated by the experimental results with our approach achieving state-of-the-art performance on both datasets.[1]

1 Introduction

Given two entities in a sentence, relation extraction (RE) extracts the relation between them and thus serves as an important task in natural language processing (NLP). Recent neural approaches for RE (Zeng et al., 2014; Zhang and Wang, 2015; Xu et al., 2015; dos Santos et al., 2015; Zhang et al., 2015; Wang et al., 2016; Zhou et al., 2016; Zhang et al., 2017) with powerful encoders (e.g., Transformers) have shown outstanding performance on benchmark datasets because the encoders are superior in capturing contextual information and thus obtain a deep understanding of the running text.

Sentence	*The*	*food*	*factory*	*produces*	*ice*	*cream*
Supertags	*NP/N*	*N/N*	*N*	*(S\NP)/NP*	*NP/N*	*N*

Figure 1: An example sentence with the CCG supertags of all words, where the supertag “*(S\NP/NP)*” of “*produces*” provides important cues to predict the relation between two given named entities “*food factory*” and “*ice cream*” (highlighted in red).

To further improve model performance, extra knowledge resources, especially the syntactic information, have been widely used for RE and demonstrated to be effective, because they provide structure information that is helpful for text understanding (Miwa and Bansal, 2016; Zhang et al., 2018; Sun et al., 2020; Chen et al., 2020). Specifically, existing approaches mainly focus on dependencies among words while paying limited attention to other types of syntactic structure, such as combinatory categorial grammar (CCG). As an important part in the a lexicalized grammatical formalism, the CCG supertags provide the lexical category of the associated words, which provides both syntactic and semantic knowledge for text understanding and thus is potentially beneficial for RE. Figure 1 shows a typical example. Herein, the supertag of “*produces*” (which is “*(S\NP/NP)*”) indicates the predicate requires to nominal arguments and the supertags of the two given entities (which are highlighted in red) suggests that they could serve as good candidates. Therefore, the supertags suggest that “*produces*” contributes more to extracting the relation between the two entities and thus guide a model to make a correct prediction.

In this paper, we propose to leverage CCG supertags to detect the relation between entities. In doing so, we use an existing CCG supertager to annotate the supertags of the input text and then run a multi-task learning process to learn from human-annotated RE and auto-annotated supertags, where

[†] Corresponding author.

[1] Our code related to this paper is available at `https://github.com/synlp/RE-CCG`.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5780–5786
December 7-11, 2022 ©2022 Association for Computational Linguistics

an attention mechanism is performed over all input words to distinguish the important ones for RE with the attention weights guided by the supertag decoding process. Therefore, our model is able to learn CCG information through supertag decoding rather than using the supertags as input features, which allows our approach to run efficiently in inference. Experimental results on two English benchmark datasets for RE, i.e., ACE2005EN and SemEval 2010 Task 8, demonstrate the effectiveness of our approach, where our approach outperforms strong baselines and achieve state-of-the-art performance on both datasets.

2 Preliminaries

RE is conventionally regarded as a text classification task with the given input sentence (which is denoted as $\mathcal{X} = x_1, \cdots, x_n$) and two entities (which are denoted as E_1 and E_2) in it, which can be formalized as

$$\widehat{y} = \underset{y \in \mathcal{T}}{\arg\max}\ p\left(y|\mathcal{X}, E_1, E_2)\right) \qquad (1)$$

where p computes the probability of the relation label $y \in \mathcal{T}$ ($\mathcal{T}$ is the label set) and $\widehat{y}$ is the model prediction. In doing so, special tokens, i.e., "*<e1>*" and "*</e1>*" for E_1 and "*<e2>*" and "*</e2>*" for E_2, are firstly inserted around the entities to mark their positions. Next, the sentence (with the special entity markers) is fed into an encoder, where the obtained hidden vectors for the i-th word x_i is denoted as $\mathbf{h}_i$. Third, the hidden vectors of the words belonging to a particular entity (i.e., $E_j, j = 1, 2$) are extracted and fed to multi-layer perceptron (MLP) for further encoding, where the resulting vectors are passed through a max pooling layer to obtain the entity representation $\mathbf{o}_j$:

$$\mathbf{o}_j = \text{MaxPooling}(\{\text{MLP}\left(\mathbf{h}_i\right)|x_i \in E_j\}) \qquad (2)$$

Then, we concatenate the entity representations $\mathbf{o} = \mathbf{o}_1 \oplus \mathbf{o}_2$ and fed the resulting $\mathbf{o}$ into a softmax classifier to predict the relation $\widehat{y}$.

3 The Proposed Approach

To leverage the information carried by CCG supertags, one strightforward approach is to use an off-the-shelf CCG supertager to annotate the supertags of each input word and then use them as extra word-level features by concatenating them with the input words before sending them to the text encoder. However, such approach requires the CCG supertagging as a pre-processing step in inference, which is not efficient especially when the data to be processed is relative large. Considering multi-task learning serves as an effective approach to learn from different tasks and it does not require the label from different tasks as extra input, we propose to learn the CCG information through a multi-task learning process and then use the CCG information to guide RE though an attention mechanism over all input words.

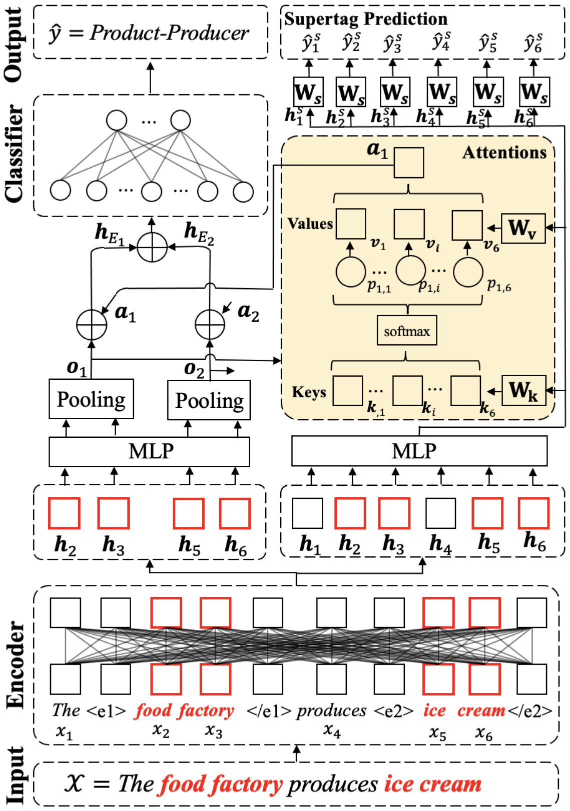
Output
$\hat{y}$ = Product-Producer
Supertag Prediction
$\hat{y}^s_1$ $\hat{y}^s_2$ $\hat{y}^s_3$ $\hat{y}^s_4$ $\hat{y}^s_5$ $\hat{y}^s_6$
W_s
h^s_1 h^s_2 h^s_3 h^s_4 h^s_5 h^s_6
Classifier
h_{E_1} h_{E_2}
a_1 a_2
o_1 o_2
Attentions
Values
W_v
v_1 v_i v_6
$p_{1,1}$ $p_{1,i}$ $p_{1,6}$
softmax
Keys
W_k
k_1 k_i k_6
Pooling
MLP
h_2 h_3 h_5 h_6
h_1 h_2 h_3 h_4 h_5 h_6
Encoder
The <e1> food factory </e1> produces <e2> ice cream</e2>
x_1 x_2 x_3 x_4 x_5 x_6
Input
$\mathcal{X}$ = The food factory produces ice cream

Figure 2: The overall architecture of the proposed approach for RE with CCG supertag guided attentions as the enhancement. The entities are highlighted in red.

The overall architecture of our model is illustrated in Figure 2, where the backbone model for RE following the standard process illustrated on the left and the CCG supertag decoding process as well as the attention mechanism illustrated on the top right. For CCG supertag decoding, we firstly take the hidden vector $\mathbf{h}_i$ of the word x_i obtained from the encoder and pass it through a MLP:

$$\mathbf{h}^s_i = \text{MLP}\left(\mathbf{h}_i\right) \qquad (3)$$

where the obtained $\mathbf{h}^s_i$ is mapped to the CCG supertag output space by a trainable matrix $\mathbf{W}_s$ and then a softmax classifier is applied to predict the supertag $\widehat{y}^s_i$ annotated by an existing supertagger. Simultaneously, $\mathbf{h}^s_i$, as well as the entity representation $\mathbf{o}_j$ obtained from the backbone model, is fed

Datasets		Sent. #	Token #	Instance #
ACE05	Train	7K	145K	5K
	Dev	2K	36K	1K
	Test	2K	31K	1K
SemEval	Train	8K	141K	8K
	Test	3K	48K	3K

Table 1: The statistics of the two English benchmark datasets used in our experiments for relation extraction, where the number of sentence, tokens, and instances (i.e., entity pairs) are reported.

Learning Rate	$5e-6, 1e-5, 2e-5, \mathbf{3e-5}$
Warmup Rate	$\mathbf{0.06}, 0.1$
Dropout Rate	$\mathbf{0.1}$
Batch Size	$16, \mathbf{32}, 64, 128$

Table 2: The hyper-parameters tested in tuning our models. The best ones used in our final experiments are highlighted in boldface.

into an attention module to enhance the RE prediction process. Specifically, we use two trainable matrix $\mathbf{W}_k$ and $\mathbf{W}_v$ to map $\mathbf{h}_i^s$ to the key vector $\mathbf{k}_i$ and value vector $\mathbf{v}_i$, respectively.

$$\mathbf{k}_i = \mathbf{W}_k \cdot \mathbf{h}_i^s, \quad \mathbf{v}_i = \mathbf{W}_v \cdot \mathbf{h}_i^s \tag{4}$$

Then, for entity E_j, we compute the attention weight $p_{j,i}$ assigned to the value $\mathbf{v}_i$ through

$$p_{j,i} = \frac{\exp\left(\mathbf{o}_j \cdot \mathbf{k}_i\right)}{\sum_{i=1}^{n} \exp\left(\mathbf{o}_j \cdot \mathbf{k}_i\right)} \tag{5}$$

Afterwards, we apply $p_{j,i}$ to the value vector $\mathbf{v}_i$ and obtain the weighted sum vector $\mathbf{a}_j$ via

$$\mathbf{a}_j = \sum_{i=1}^{n} p_{j,i} \cdot \mathbf{v}_i \tag{6}$$

Finally, we concatenate $\mathbf{a}_j$ with the entity representation $\mathbf{o}_j$ to obtain the enhanced entity representation $\mathbf{h}_{E_j} = \mathbf{o}_j \oplus \mathbf{a}_j$. Once the enhanced representations of the two entities are computed, we concatenate them and feed the resulting vector to the softmax classifier, following the standard RE decoding process.

In training, the model is optimized on RE and CCG supertagging, which allows our model to learn CCG information and use it to enhance the entity representation through the attention mechanism with the attention weights assigned to different input words guided by the learnt CCG information.

Models	ACE05		SemEval	Para. #	Speed
	Dev	Test			
BERT	76.11	76.94	89.03	335M	17.1
+ GCN	78.45	78.56	89.38	336M	15.0
+ GAT	78.77	78.80	89.47	336M	14.6
+ Ours	**79.15**	**79.20**	**89.88**	336M	17.0

Table 3: Experimental results of the BERT-large baseline, GCN, GAT, and our approach on the development and test sets of ACE05 and SemEval, where model size (i.e., the number of model parameters) and the inference speed (i.e., the number of processed sentence per second) are also reported.

4 Experiments

4.1 Settings

Following previous studies (Hendrickx et al., 2010; Zeng et al., 2014; Zhang and Wang, 2015; Xu et al., 2015; Zhou et al., 2016; Zhang et al., 2017; Soares et al., 2019; Qin et al., 2021; Chen et al., 2021; Tian et al., 2022), we use two English benchmark datasets for RE to evaluate the proposed model. The first is ACE2005EN (ACE05)[2], where the English section is used in the experiments and two small subset of relation types, namely cts and un are removed following the convention in previous studies (Miwa and Bansal, 2016; Christopoulou et al., 2018; Ye et al., 2019; Tian et al., 2021). ACE05 is split into training, development, and test set according to Miwa and Bansal (2016)[3]. The second dataset is SemEval 2010 Task 8 (SemEval) (Hendrickx et al., 2010), where we use the official training and test split (SemEval does not have an official development set). Table 1 reports the statistics of the datasets.

We try two graph-based models as our baseline for comparison, namely, graph convolutional networks (GCN) (Kipf and Welling, 2016) and graph attentive networks (GAT) (Veličković et al., 2017). We use the dependency tree obtained through Stanford CoreNLP Toolkits (Manning et al., 2014) to build the word graph and use the graph as additional input to the GCN and GAT models.

We use the CCG supertager[4] proposed by Tian et al. (2020b) to annotate the CCG supertags for multi-task learning. For the encoder, consider a high-quality text representation plays an important

[2] We obtain the official data (LDC2006T06) from `https://catalog.ldc.upenn.edu/LDC2006T06`.

[3] We follow the train/dev/test splits specified by Miwa and Bansal (2016) at `https://github.com/tticoin/LSTM-ER/tree/master/data/ace2005/split`

[4] `https://github.com/cuhksz-nlp/NeST-CCG`

Models	ACE05	SemEval
Socher et al. (2012)	-	82.4
Zeng et al. (2014)	-	82.7
Zhang and Wang (2015)	-	79.6
Xu et al. (2015)	-	83.7
Wang et al. (2016)	-	88.0
Zhou et al. (2016)	-	84.0
†Zhang et al. (2018)	-	84.8
Wu and He (2019)	-	89.2
Christopoulou et al. (2018)	64.2	-
Ye et al. (2019)	68.9	-
†Guo et al. (2019)	-	85.4
Baldini Soares et al. (2019)	-	89.5
†Mandya et al. (2020)	-	85.9
†Sun et al. (2020)	-	86.0
†Yu et al. (2020)	-	86.4
Wang et al. (2020)	66.7	-
Wang and Lu (2020)	67.6	-
Wang et al. (2021)	66.0	-
†Tian et al. (2021)	79.05	89.85
†Ours	**79.10**	**89.96**

Table 4: The comparison of F1 scores between previous studies and our best model with BERT-large on the test sets of ACE05 and SemEval. Previous studies that leverage syntactic information (e.g., the dependency tree of the input sentence) are marked by "†".

role to achieve good model performance in downstream NLP tasks (Song and Shi, 2018; Han et al., 2018; Devlin et al., 2019; Radford et al., 2019; Tian et al., 2020a; Lewis et al., 2020; Diao et al., 2020; Raffel et al., 2020; Diao et al., 2021; Song et al., 2021), we try the large version of BERT[5] (Devlin et al., 2019) (which achieves state-of-the-art performance in many NLP tasks) with the default settings (i.e., 24 layers of multi-head attentions with 1024-dimensional hidden vectors). For evaluation, we follow previous studies to use the standard micro-F1 scores[6] for ACE05 and use the macro-averaged F1 scores[7] for SemEval. In our experiments, we try different combinations of hyper-parameters (which are illustrated in Table 2 with the best ones highlighted in boldface), and tune them on the dev set, then evaluate on the test set by the model that achieves the highest F1 score on the dev set.

4.2 Results

Table 3 shows the average[8] F1 scores of different models (including the vanilla BERT-large baseline, the GCN and GAT baseline, and our approach) on the development and test set of ACE05 and SemEval, where the size of different models (in terms of the number of parameters) and the inference speed (in terms of the number of processed sentences per second) are also reported for reference.

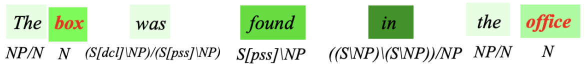

Figure 3: Visualizations of weights assigned to different words for an example input sentence, where the supertags associated with them are illustrated at the bottom. Darker background color refer to higher weights.

There are several observations. First, our model works well with the BERT-large pre-trained language model, where the consistent improvement is observed over the vanilla BERT baselines on both datasets, although the BERT baselines have already achieve outstanding performance. Second, it is promising to observe that our model outperforms the standard GCN and GAT that leverage dependencies on both datasets, which further confirms the effectiveness of our approach. We attribute this observation to the superior of CCG supertag that carries both syntactic and semantic information of the running text and thus is able to provide a deeper analysis of the text and use it to guide the relation prediction process. Third, it is observed that our model is able to perform more efficient compared with GCN and GAT, because the CCG information is learnt through training in our approach and no supertags is required as input in inference whereas GCN and GAT require the input to be parsed before they can predict the relation.

We further compare our approach with recent previous studies and report the results in Table 4. It is promising to observe that our approach outperforms previous studies (including the ones with powerful encoder and syntactic information) and achieves state-of-the-art performance on both datasets, which further confirms the effectiveness of our approach.

4.3 Case Study

To illustrate how CCG information guide the relation extraction process through the attention mechanism, in Figure 3, we visualize the average attention weights assigned to different words in an example sentence (the entities are highlighted in red and their gold standard relation is "*location*") by word background color, where higher weights correspond to deeper color. The CCG supertags of the words are shown below the attached words. It is

[5] We download pre-trained BERT-large model from `https://github.com/huggingface/transformers`.

[6] We use the evaluation script from *sklearn* framework.

[7] We use the official evaluation script downloaded from `http://semeval2.fbk.eu/scorers/task08/SemEval2010_task8_scorer-v1.2.zip`.

[8] For each model, we run it five times with different random seeds and report the average performance.

worthnoting that the supertags are given for better illustration; they are not used as input in inference. In this case, our model is able to distinguish that "*in*" tend to be the head of a prepositional phrase (PP) that is attached to a predicate based on the learnt CCG supertag information[9] and its argument noun phrase is exactly one of the given entities. Therefore, our model assigns the highest weight to "*in*", which strongly suggests a "*location*" relation, and thus results in the correct relation prediction.

5 Conclusion

In this paper, we propose a neural approach for improving RE through a CCG guided attention mechanism, where our model learns the CCG information through a multi-task learning process to predict RE and CCG supertags simultaneously and uses the learnt CCG information to compute the attention weights assigned to different words. In doing so, our approach is able to learn the CCG information through CCG supertag decoding rather than using it as additional input features, which allows our model to run efficiently in inference. Experimental results on two English benchmark datasets (i.e., ACE05 and SemEval) for RE demonstrate the effectiveness of our approach, where state-of-the-art performance is obtained on both datasets.

References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.

Guimin Chen, Yuanhe Tian, and Yan Song. 2020. Joint Aspect Extraction and Sentiment Analysis with Directional Graph Convolutional Networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 272–279.

Guimin Chen, Yuanhe Tian, Yan Song, and Xiang Wan. 2021. Relation Extraction with Type-aware Map Memories of Word Dependencies. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2501–2512, Online.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2018. A Walk-based Model on Entity Graphs for Relation Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 81–88.

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.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2020. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4729–4740.

Shizhe Diao, Ruijia Xu, Hongjin Su, Yilei Jiang, Yan Song, and Tong Zhang. 2021. Taming Pre-trained Language Models with N-gram Representations for Low-Resource Domain Adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3336–3349, Online.

Cícero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying Relations by Ranking with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251.

Jialong Han, Yan Song, Wayne Xin Zhao, Shuming Shi, and Haisong Zhang. 2018. Hyperdoc2vec: Distributed Representations of Hypertext Documents. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2384–2394, Melbourne, Australia.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38.

Thomas N Kipf and Max Welling. 2016. Semi-supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online.

[9] The supertag of "*in*" is more likely to be "*(N\N)/NP*" if it is attached to an noun phrase.

Angrosh Mandya, Danushka Bollegala, and Frans Coenen. 2020. Graph Convolution over Multiple Dependency Sub-graphs for Relation Extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6424–6435.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

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.

Han Qin, Yuanhe Tian, and Yan Song. 2021. Relation Extraction with Word Graphs from N-grams. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2860–2868, Online and Punta Cana, Dominican Republic.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-text Transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic Compositionality through Recursive Matrix-Vector Spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211.

Yan Song and Shuming Shi. 2018. Complementary Learning of Word Embeddings. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4368–4374.

Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. 2021. ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders. *arXiv preprint arXiv:2105.01279*.

Kai Sun, Richong Zhang, Yongyi Mao, Samuel Mensah, and Xudong Liu. 2020. Relation Extraction with Convolutional Network over Learnable Syntax-Transport Graph. In *AAAI*, pages 8928–8935.

Yuanhe Tian, Guimin Chen, Yan Song, and Xiang Wan. 2021. Dependency-driven Relation Extraction with Attentive Graph Convolutional Networks. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Yuanhe Tian, Yan Song, and Fei Xia. 2020a. Joint Chinese Word Segmentation and Part-of-speech Tagging via Multi-channel Attention of Character N-grams. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2073–2084.

Yuanhe Tian, Yan Song, and Fei Xia. 2020b. Supertagging Combinatory Categorial Grammar with Attentive Graph Convolutional Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6037–6044.

Yuanhe Tian, Yan Song, and Fei Xia. 2022. Improving Relation Extraction through Syntax-induced Pre-training with Dependency Masking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph Attention Networks. *arXiv preprint arXiv:1710.10903*.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online.

Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. 2016. Relation Classification via Multi-Level Attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Junchi Yan, Peng Gao, and Guotong Xie. 2020. Pre-training entity relation encoder with intra-span and inter-span information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1692–1705, Online.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021. UNIRE: A Unified Label Space for Entity Relation Extraction. *arXiv preprint arXiv:2107.04292*.

Shanchan Wu and Yifan He. 2019. Enriching Pre-trained Language Model with Entity Information for Relation Classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2361–2364.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. 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*, pages 1785–1794.

Wei Ye, Bo Li, Rui Xie, Zhonghao Sheng, Long Chen, and Shikun Zhang. 2019. Exploiting Entity BIO Tag Embeddings and Multi-task Learning for Relation Extraction with Imbalanced Data. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1351–1360.

Bowen Yu, Mengge Xue, Zhenyu Zhang, Tingwen Liu, Wang Yubin, and Bin Wang. 2020. Learning to Prune Dependency Trees with Rethinking for Neural Relation Extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3842–3852, Barcelona, Spain (Online).

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation Classification via Convolutional Deep Neural Network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78.

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.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212.