Dependency-aware Prototype Learning for Few-shot Relation Classification

Tianshu Yu [1,2], **Min Yang** [1,*], **Xiaoyan Zhao** [1]
[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
ts.yu@siat.ac.cn, min.yang@siat.ac.cn, xy.zhao@siat.ac.cn

Abstract

Few-shot relation classification aims to classify the relation type between two given entities in a sentence by training with a few labeled instances for each relation. However, most of existing models fail to distinguish multiple relations that co-exist in one sentence. This paper presents a novel dependency-aware prototype learning (DAPL) method for few-shot relation classification. Concretely, we utilize dependency trees and shortest dependency paths (SDP) as structural information to complement the contextualized representations of input sentences by using the dependency-aware embedding as attention inputs to learn attentive sentence representations. In addition, we introduce a gate controlled update mechanism to update the dependency-aware representations according to the output of each network layer. Extensive experiments on the FewRel dataset show that DAPL achieves substantially better performance than strong baselines. For reproducibility, we will release our code and data upon the publication of this paper at https://github.com/publicstaticvo/DAPL.

1 Introduction

Relation classification, which aims to classify the relation between two entities in a sentence, is a fundamental task for information retrieval (Kadry and Dietz, 2017), knowledge graph construction (Shen et al., 2020; Ji et al., 2021) and question answering (Luo et al., 2018). Most of existing relation classification methods (Wang et al., 2016; Guo et al., 2019; Shen et al., 2020; Tian et al., 2021; Zhao et al., 2022a) focus on the supervised scenario where sufficient labeled training data is available. However, it is time-consuming and labor-intensive to collect large-scale labeled data in many real-world applications, especially in the low-resource settings (Geng et al., 2019, 2020; Fan et al., 2021; Zhao et al., 2022b).

Recently, few-shot relation classification (FSRC), which explores relation extraction methods by training with a few labeled examples in each relation, has become a hot research topic (Gao et al., 2019; Qu et al., 2020; Gao et al., 2020; Wang et al., 2020; Xu and Xiang, 2021; Ding et al., 2021; Fan et al., 2021). For instance, Han et al. (2018) introduce a large-scale FSRC dataset and implement several well-known few-shot learning techniques (Finn et al., 2017; Snell et al., 2017) for FSRC. Qu et al. (2020) propose a Bayesian meta learning approach for FSRC, which learns the posterior distributions of prototype vectors among different relations.

Despite the remarkable progress of FSRC methods, there is still a technical challenge which is not addressed well in prior work. Specifically, there can be multiple relations that co-exist in a sentence, while only one relation corresponds to the given entity pairs. The other existed relations may mislead the classifier to the wrong relation class, which is called the *misleading relation*. Taking Figure 1 as an example, the gold relation between two target entities "*Mitsubishi toppo*" and "*minica*" is "*derivative-model*" marked by the term "*derived from*", while most prior FSRC methods incorrectly predict the misleading relation "*products-producer*" marked by the term "*produced by*".

One possible solution is to leverage the dependency tree as auxiliary information to facilitate the representation learning. Recently, several studies have incorporated dependency tree into supervised relation classification models and obtained significant performance improvement (Sun et al., 2020; Yu et al., 2020; Pouran Ben Veyseh et al., 2020; Chen et al., 2021; Fan et al., 2021; Tian et al., 2021). However, few studies investigate the effectiveness of dependency trees in FSRC task. In addition, most existing works either solely focus on the terms that have direct dependency with target entities or involve redundant information by using

*Min Yang is corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 2339–2345
October 12–17, 2022.

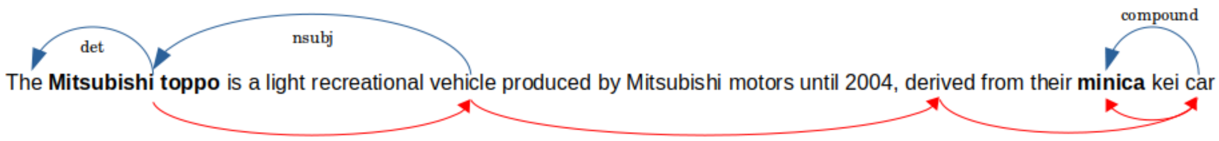

Figure 1: An example from the test set of FewRel. Previous models only focus on the dependency tree in blue color and ignore the SDP in red color which entails the ground-truth relation.

the entire dependency tree, failing to get other information such as shortest dependency paths (SDP) of two entities thus cannot tackle the misleading relation problem. For example, as illustrated in Figure 1, the SDP (marked as red lines) of the two entities can help the relation classification model obtain the correct relation "*derived from*". Therefore, it is necessary to fully exploit dependency information as auxiliary structural information, which can help identify useful terms and misleading terms via their relative positions to the given entities.

In this paper, we propose a novel dependency-aware prototype learning (DAPL) method for FSRC. DAPL is based on the framework of prototypical networks (Snell et al., 2017) with the BERT (Devlin et al., 2018) encoder, motivated by the effectiveness of prototypical networks in few-shot classification tasks. In our method, we leverage dependency trees as structural information to complement the contextualized representations of input sentences. Specifically, we assign each input token with a dependency label, according to whether the token is adjacent to the target entities or on the SDP between the two target entities. We highlight the tokens on the SDP by assigning a unified sdp dependency label for each token. Then, we convert these dependency labels into dependency embeddings, which are used as attention inputs of the contextualized sentence representations to learn dependency-aware sentence representations. Furthermore, we introduce a gate-controlled update mechanism to update the dependency-aware representations based on the output of each BERT layer, inspired by the effectiveness of the gate update functions in GRU (Cho et al., 2014). This mechanism provides a feedback to dependency states about whether they are reflecting the importance of each token and related to the ground truth labels.

The main contributions of this work are threefold:

- We propose a novel dependency-aware prototype learning method for FSRC, which fully exploit the dependency and contextualized information to alleviate the misleading relation problem and improve the overall performance of FSRC.
- We present a gate-controlled update mechanism to adaptively adjust the dependency-aware representations according to the output of each network layer.
- Experiments on a benchmark FSRC dataset (i.e., FewRel) show that our method outperforms the strong baselines by a noticeable margin.

2 Methodology

Problem Definition In the RC task, each instance consists of an input sequence x (including a input sentence z, a head entity e_1, a tail entity e_2) and a relation label y for the two entities. We adopt a typical N-way-K-shot setting for FSRC (Qu et al., 2020). Under N-way-K-shot configuration, the training data is further split into a support set $\mathcal{S}$ and a query set $\mathcal{Q}$ which have disjoint labels, where $\mathcal{S}$ contains N relation classes and each with K labeled examples. The goal of FSRC is to learn a model using D_{train}, which is then used to predict the relation y for each input x in testing set.

2.1 Dependency Labels

Given an input example $x \in D_{\text{train}}$, we denote its dependency tree as $\mathcal{G} = (V, E)$, where V contains the tokens in the sentence and E contains the set of edges (dependencies) of tokens. Each triplet $(w_i, w_j, d) \in E$ denotes that there is a dependency of type d between tokens w_i and w_j in x. Note that $\mathcal{G}$ is an undirected graph. Given the head entity e_1 and the tail entity e_2, we denote the set of all tokens on the SDP between e_1 and e_2 except themselves as P. We assign two dependency labels $l_i^{(1)}$ and $l_i^{(2)}$ to each token w_i of the sentence x, where $l_i^{(1)}$ and $l_i^{(2)}$ denote the dependency relations between the token w_i to the head entity and the tail entity respectively by the following four steps:

1. We initialize the $l_i^{(1)}$ and $l_i^{(2)}$ labels of each token as $none$.
2. The $l_i^{(1)}$ label of e_1 and the $l_i^{(2)}$ label of e_2 are set to $self$.
3. For each token $w_i \in P$ on SDP except e_1 and e_2, we set its $l_i^{(1)}$ and $l_i^{(2)}$ labels as sdp.
4. For each token $w_i \notin P$ that is not on SDP, we set $l_i^{(1)}$ to the corresponding dependency parsing type if $l_i^{(1)}$ is $none$ and w_i has an edge connected to e_1 on the dependency tree. We can get the $l_i^{(2)}$ label for e_2 in a similar way.

To better illustrate the above process, we take the sentence "*[CLS] the school <e1> master </e1> teaches the lesson with a <e2> steak </e2> [SEP]*" as an example. We show how the two labels of each token are obtained as follows:

1. We initialize the $l_i^{(1)}$ and $l_i^{(2)}$ labels of each token as $none$.
2. The $l_i^{(1)}$ labels of "<e1>", "master", "</e1>" and the $l_i^{(2)}$ labels of "<e2>", "steak", "</e2>" are assigned with $self$.
3. The dependency path between the two entities (i.e., "master" and "steak") is "master-teaches-steak", so both $l_i^{(1)}$ and $l_i^{(2)}$ labels of "teaches" are set as sdp.
4. For the remaining tokens, "the" and "school" are adjacent to "master" on the dependency tree, so the $l_i^{(1)}$ label of "the" is det, and the $l_i^{(1)}$ label of "school" is $compound$. Meanwhile, "with" and "a" are adjacent to "steak", so the $l_i^{(2)}$ label of "with" is $case$, and the $l_i^{(2)}$ label of "a" is det.

Afterwards, we use an embedding layer to convert the dependency labels $l_i^{(1)}$ and $l_i^{(2)}$ into dependency embeddings $\mathbf{d}_i^{(1)}$ and $\mathbf{d}_i^{(2)}$ with an embedding dimension of $d_h/2$, where d_h is the hidden vector size of the encoder. The dependency embedding $\mathbf{d}_i$ of each token w_i is formed by concatenating $\mathbf{d}_i^{(1)}$ and $\mathbf{d}_i^{(2)}$ together, whose length is d_h.

2.2 Dependency-aware Attention

Figure 2 shows the structure of our model DAPL. Our model takes each token representation $\{\mathbf{w}_i\}$ and dependency embedding $\{\mathbf{d}_i\}$ in the sentence

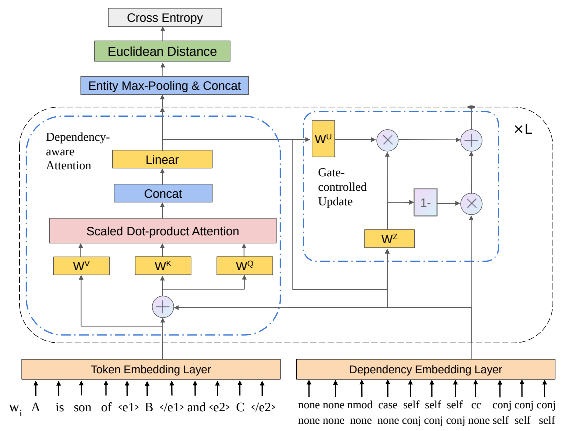

Figure 2: The overall structure of our DAPL.

as input. Inspired by the remarkable success of pre-trained language models (PLMs) on most of NLP tasks, we employ BERT (Devlin et al., 2018) as the basic framework of our model. To learn the importance of each token to the given entities, we modify the self-attention mechanism in original BERT by adding together the contextual representation and dependency representation when generating query and key matrices at the l-th layer:

$$Q^{(l)} = (\mathbf{h}_i^{(l-1)} + \mathbf{d}_i^{(l-1)})W_Q^{(l)} \tag{1}$$

$$K^{(l)} = (\mathbf{h}_i^{(l-1)} + \mathbf{d}_i^{(l-1)})W_K^{(l)} \tag{2}$$

$$V^{(l)} = \mathbf{h}_i^{(l-1)}W_V^{(l)} \tag{3}$$

$$\tilde{\mathbf{h}}^{(l)} = \mathrm{softmax}\left(\frac{Q^{(l)}K^{(l)T}}{\sqrt{d_K}}\right)V^{(l)} \tag{4}$$

where $W_Q^{(l)}, W_K^{(l)}, W_V^{(l)} \in \mathbb{R}^{d_h \times d_h}$ are learnable attention weights in scaled dot-product attention. Here, $\mathbf{h}_i^{(0)} = \mathbf{w}_i$ and $\mathbf{d}_i^{(0)} = \mathbf{d}_i$. Then, a two-layer feed-forward neural network with a ReLU activation takes the weighted sum $\tilde{\mathbf{h}}^{(l)}$ as input to learn the output hidden states $\mathbf{h}^{(l)}$ at the l-th layer:

$$\mathbf{h}^{(l)} = \max(0, \tilde{\mathbf{h}}^{(l)}W_1^{(l)} + \mathbf{b}_1^{(l)})W_2^{(l)} + \mathbf{b}_2^{(l)} \tag{5}$$

where $W_1^{(l)}$, $W_2^{(l)}$, $\mathbf{b}_1^{(l)}$, $\mathbf{b}_2^{(l)}$ are learnable parameters in BERT.

2.3 Gate-controlled Update

We propose a gate-controlled update to the dependency states $\mathbf{d}_i^{(l-1)}$ at the end of each layer by using the previous dependency representation $\mathbf{d}_i^{(l-1)}$ and the output hidden states $\mathbf{h}_i^{(l)}$. Inspired by the Gate Recurrent Unit (GRU) (Cho et al., 2014), we devise an update gate and a control gate. Specifically, the control gate is a single fully-connected layer with a sigmoid activate function, which is defined as:

$$\mathbf{z}_i^{(l)} = \text{sigmoid}([\mathbf{h}_i^{(l)}; \mathbf{d}_i^{(l-1)}]W_Z^{(l)}) \quad (6)$$

where $W_Z^{(l)} \in \mathbb{R}^{2d_h \times d_h}$ is a learnable parameter. The update gate is a single fully-connected layer with a tanh activate function, which is defined as:

$$\mathbf{u}_i^{(l)} = \tanh(\mathbf{h}_i^{(l)} W_U^{(l)}) \quad (7)$$

where $W_U^{(l)} \in \mathbb{R}^{d_h \times d_h}$ is a learnable parameter.

Finally, the output dependency representations are learned by considering the last dependency state $\mathbf{d}_i^{(l-1)}$ and the update gate output $\mathbf{u}_i^{(l)}$ under the control of $\mathbf{z}_i^{(l)}$:

$$\mathbf{d}_i^{(l)} = (1 - \mathbf{z}_i^l) \odot \mathbf{d}_i^{(l-1)} + \mathbf{z}_i^l \odot \mathbf{u}_i^{(l)} \quad (8)$$

where $\odot$ represents the element-wise product.

2.4 Relation Classification

We apply a max-pooling operation on the position spans of the head and tail entities, and get the head entity representation $\mathbf{h}_{e_1}^L$ and tail entity representation $\mathbf{h}_{e_2}^L$, where L denotes the number of layers in BERT. Then, we concatenate $\mathbf{h}_{e_1}$ and $\mathbf{h}_{e_2}$ as the representation $\mathbf{h}$ of each input instance.

Following the prototypical network (Snell et al., 2017), we compute a prototype for each relation class c as $\mathbf{p}_c = \frac{1}{K_c}\sum_{(x_{s_i}, y_{s_i}) \in \mathcal{S}_c} \mathbf{h}_{x_{s_i}}$, where $\mathcal{S}_c = \{(x_{s_i}, y_{s_i})\}_{i=1}^{K_c}$ denotes the support set that has class label c, $\mathbf{h}_{x_{s_i}}$ is the contextual representation of x_{s_i}, and K_c is the number of instances in $\mathcal{S}_c$. Given the query set $\mathcal{Q} = \{(x_{q_i}, y_{q_i})\}_{i=1}^{K_\mathcal{Q}}$ and a Euclidean distance function $d(\cdot)$, the prototypical network computes a distribution over classes for a query instance x_{q_i} based on a softmax over distances to the prototypes in the embedding space. Formally, we define the prototypical objective $\mathcal{L}_{\text{proto}}$ over the query set $\mathcal{Q}$ as follows:

$$\mathcal{L}_{\text{proto}} = -\frac{1}{K_\mathcal{Q}} \sum_{i=1}^{K_\mathcal{Q}} \log \frac{\exp(-d(\mathbf{h}_{x_{q_i}}, \mathbf{p}_{y_{q_i}}))}{\sum_{c=1}^{N} \exp(-d(\mathbf{h}_{x_{q_i}}, \mathbf{p}_c))} \quad (9)$$

where $K_\mathcal{Q}$ denotes the number of instances in $\mathcal{Q}$.

Inference Stage In inference phase, we compute the relation $\hat{y}_i$ of each input x_i in testing set as:

$$\hat{y}_i = \operatorname*{argmin}_c d(\mathbf{h}_{x_i}, \mathbf{p}_c), \ c \in [1, \ldots, N] \quad (10)$$

3 Experiments

3.1 Experimental Setup

Dataset We use the benchmark FSRC dataset FewRel (Han et al., 2018) to evaluate the effectiveness of our model. FewRel contains 100 different relations, with 64 relations in training set, 16 relations in validation set and 20 relations in testing set. For each type of relation, there are 700 different examples. Since the 20 testing relations are unpublished, we re-split the published 64 training relations into 50 relations and 14 relations for training and validation respectively, and employ the original validation set with 16 relations for testing, following previous studies (Yang et al., 2020).

Baselines We compare DAPL with several state-of-the-art baselines for FSRC, including **Proto** (Snell et al., 2017), **Proto-GAT** (Snell et al., 2017), **BERT-PAIR** (Gao et al., 2019), **CTEG** (Wang et al., 2020), **TD-Proto** (Yang et al., 2020), and a simple version of **ConceptFERE** (Yang et al., 2021) that involves an external concept database.

Implementation Details For the PLM, the proposed DAPL is implemented based on BERT$_{base}$ for all experiments. We conduct N-way-K-shot (denoted as N-w-K-s) to study the performance in different situations. Here, we adopt four different settings, i.e., 5-w-1-s, 5-w-5-s, 10-w-1-s, and 10-w-5-s. We tune the entire model and select the checkpoint with best validation performance. The maximum length of the sentence is 90. We follow Soares et al. (2019) to insert four special tokens <e1>, </e1>, <e2> and </e2> to mark the boundaries of the entities. The dependency trees are obtained using the external Standard CoreNLP Toolkit proposed by StanfordNLP. The size of the dependency embedding is 384. DAPL is optimized with AdamW (Loshchilov and Hutter, 2018) and warmup mechanism (Popel and Bojar, 2018).

3.2 Experimental Results

Overall Results We adopt classification accuracy as the evaluation metric. Table 1 reports the experimental results of our model and four baselines in four few-shot settings. Our DAPL model achieves significantly better performance than the baselines in all settings. Specifically, our method improves the best performance of baselines by 0.28%/1.76%/0.75%/3.34% under the four settings respectively. The performance gain of our method comes from the auxiliary dependency information.

Model	5 way 1 shot	5 way 5 shot	10 way 1 shot	10 way 5 shot
Proto	72.65	86.15	60.13	76.20
Proto-GAT	79.14	88.46	68.87	79.45
BERT-PAIR	85.66	89.48	76.84	81.76
ConceptFERE	84.28	90.34	74.00	81.82
CTEG	84.72	92.52	76.01	84.89
TD-Proto	84.76	92.38	74.32	85.92
DAPL (Ours)	**85.94**	**94.28**	**77.59**	**89.26**
DAPL w/o Gate	85.44	93.68	76.29	88.27
DAPL w/o SDP	85.30	93.10	76.04	87.43
DAPL w/o DT	85.06	92.46	75.13	86.54

Table 1: The main evaluation results and the ablation results on the test set.

Ground-truth	By DAPL	By CTEG	Input Example
Husband-Wife	Husband-Wife	Children-Parent	He was born in Kristiania as a son of Gerda Ring and Halfdan Christensen and brother of Bab Christensen.
Parent-Children	Parent-Children	Husband-Wife	On March 8,1852 he married Kapi'olani, daughter of Kūhiō Kalaniana'ole and Kinoiki Kekaulike.

Table 2: Prediction results on the test samples. We use the red and blue colors to indicate the head and tail entities.

Ablation Study To analyze the impact of different components in DAPL, we also conduct ablation test in terms of discarding the dependency tree (denoted as w/o DT), the SDP dependency label (denoted as w/o SDP) and the gate-controlled update mechanism (denoted as w/o Gate). The ablation test results are reported in Table 1. The accuracy scores decrease sharply when removing the dependency tree. This is within our expectation since the dependency tree provides rich information of entities and relations between them. Not surprisingly, combining all the factors achieves the best performance over the four experimental settings.

Case Study In Table 2, we provide a case study to illustrate the effectiveness of our model for alleviating the misleading relation problem qualitatively. Specifically, we provide two examples from the test set that are incorrectly predicted by CTEG while being correctly predicted by our method. By fully exploiting the auxiliary dependency information, our DAPL can correctly predict the correct relation even being disturbed by the misleading relation "*Husband-wife*". However, CTEG has a propensity to confuse the co-exist relations in a sentence, since the misleading terms are close to the given entities.

4 Conclusion

In this paper, we proposed a dependency-aware prototype learning method for FSRC, which leveraged dependency trees and shortest dependency paths as structural information to complement the contextualized representations of input sentences. A gate-controlled update mechanism was further devised to adaptively update the dependency-aware representations according to the output of each network layer. Experimental results showed that DAPL outperformed strong baselines for FSRC.

Acknowledgements

This work was partially supported by National Natural Science Foundation of China (61906185, 61876053), the Natural Science Foundation of Guangdong Province of China (No. 2019A1515011705), Youth Innovation Promotion Association of CAS China (No. 2020357), Shenzhen Science and Technology Innovation Program (Grant No. KQTD20190929172835662), Shenzhen Basic Research Foundation (No. JCYJ20210324115614039 and No. JCYJ20200109113441941).

References

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.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger

Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ning Ding, Xiaobin Wang, Yao Fu, Guangwei Xu, Rui Wang, Pengjun Xie, Ying Shen, Fei Huang, Hai-Tao Zheng, and Rui Zhang. 2021. Prototypical representation learning for relation extraction. *arXiv preprint arXiv:2103.11647*.

Siqi Fan, Binbin Zhang, Silin Zhou, Menghan Wang, and Ke Li. 2021. Few-shot relation extraction towards special interests. *Big Data Research*, 26:100273.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Tianyu Gao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2020. Neural snowball for few-shot relation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7772–7779.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. Fewrel 2.0: Towards more challenging few-shot relation classification. *arXiv preprint arXiv:1910.07124*.

Ruiying Geng, Binhua Li, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Dynamic memory induction networks for few-shot text classification. In *ACL*.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *EMNLP*.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. *arXiv preprint arXiv:1906.07510*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *arXiv preprint arXiv:1810.10147*.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Amina Kadry and Laura Dietz. 2017. Open relation extraction for support passage retrieval: Merit and open issues. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1149–1152.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Kangqi Luo, Fengli Lin, Xusheng Luo, and Kenny Zhu. 2018. Knowledge base question answering via encoding of complex query graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2185–2194.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *arXiv preprint arXiv:1804.00247*.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. Exploiting the syntax-model consistency for neural relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Meng Qu, Tianyu Gao, Louis-Pascal Xhonneux, and Jian Tang. 2020. Few-shot relation extraction via bayesian meta-learning on relation graphs. In *International Conference on Machine Learning*, pages 7867–7876. PMLR.

Ying Shen, Ning Ding, Hai-Tao Zheng, Yaliang Li, and Min Yang. 2020. Modeling relation paths for knowledge graph completion. *IEEE Transactions on Knowledge and Data Engineering*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Kai Sun, Richong Zhang, Yongyi Mao, Samuel Mensah, and Xudong Liu. 2020. Relation extraction with convolutional network over learnable syntax-transport graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 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 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4458–4471.

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.

Yingyao Wang, Junwei Bao, Guangyi Liu, Youzheng Wu, Xiaodong He, Bowen Zhou, and Tiejun Zhao. 2020. Learning to decouple relations: Few-shot relation classification with entity-guided attention and confusion-aware training. *arXiv preprint arXiv:2010.10894*.

Shiyao Xu and Yang Xiang. 2021. Frog-gnn: multi-perspective aggregation based graph neural network for few-shot text classification. *Expert Systems with Applications*, 176:114795.

Kaijia Yang, Nantao Zheng, Xinyu Dai, Liang He, Shujian Huang, and Jiajun Chen. 2020. Enhance prototypical network with text descriptions for few-shot relation classification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2273–2276.

Shan Yang, Yongfei Zhang, Guanglin Niu, Qinghua Zhao, and Shiliang Pu. 2021. Entity concept-enhanced few-shot relation extraction. *arXiv preprint arXiv:2106.02401*.

Bowen Yu, Xue Mengge, 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.

Xiaoyan Zhao, Min Yang, Qiang Qu, Ruifeng Xu, and Jieke Li. 2022a. Exploring privileged features for relation extraction with contrastive student-teacher learning. *TKDE*.

Yingxiu Zhao, Zhiliang Tian, Huaxiu Yao, Yinhe Zheng, Dongkyu Lee, Yiping Song, Jian Sun, and Nevin-L. Zhang. 2022b. Improving meta-learning for low-resource text classification and generation via memory imitation. In *ACL*.