Exploring Self-Distillation Based Relational Reasoning Training for Document-Level Relation Extraction

Liang Zhang[1,2], Jinsong Su[1,2*], Zijun Min[1,2], Zhongjian Miao[1,2], Qingguo Hu[1,2]
Biao Fu[1,2], Xiaodong Shi[1,2], Yidong Chen[1,2*]
[1]School of Informatics, Xiamen University, China
[2]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, China
lzhang@stu.xmu.edu.cn, {jssu,ydchen}@xmu.edu.cn

Abstract

Document-level relation extraction (RE) aims to extract relational triples from a document. One of its primary challenges is to predict *implicit relations* between entities, which are not explicitly expressed in the document but can usually be extracted through *relational reasoning*. Previous methods mainly implicitly model relational reasoning through the interaction among entities or entity pairs. However, they suffer from two deficiencies: 1) they often consider only one reasoning pattern, of which coverage on relational triples is limited; 2) they do not explicitly model the process of relational reasoning. In this paper, to deal with the first problem, we propose a document-level RE model with a reasoning module that contains a core unit, the *reasoning multi-head self-attention* unit. This unit is a variant of the conventional multi-head self-attention and utilizes four attention heads to model four common reasoning patterns, respectively, which can cover more relational triples than previous methods. Then, to address the second issue, we propose a self-distillation training framework, which contains two branches sharing parameters. In the first branch, we first randomly mask some entity pair feature vectors in the document, and then train our reasoning module to infer their relations by exploiting the feature information of other related entity pairs. By doing so, we can explicitly model the process of relational reasoning. However, because the additional masking operation is not used during testing, it causes an input gap between training and testing scenarios, which would hurt the model performance. To reduce this gap, we perform conventional supervised training without masking operation in the second branch and utilize Kullback-Leibler divergence loss to minimize the difference between the predictions of the two branches. Finally, we conduct comprehensive experiments on three benchmark datasets, of which experimental results demonstrate that our model consistently outperforms all competitive baselines. Our source code is available at https://github.com/DeepLearnXMU/DocRE-SD.

Introduction

Human knowledge can be efficiently expressed and stored in the form of relational triple (e_s, r, e_o), where e_s and e_o are subject and object entities, respectively, and r represents the relation between them. Since these structured knowledge could benefit many downstream applications, e.g., question answering (Dong et al. 2015) and information retrieval (Wang et al. 2017), many efforts have been devoted to the studies of relation extraction (RE), which aims at automatically extracting relational triples from plain text. In this aspect, previous studies focus primarily on sentence-level RE, where both considered subject and object entities are located within the same sentence (Zhang, Qi, and Manning 2018; Baldini Soares et al. 2019). However, the performance of sentence-level RE models are often unsatisfactory in practice since abundant of relational facts are expressed by multiple sentences (Yao et al. 2019). Therefore, many recent studies (Xu et al. 2021; Zhou et al. 2021) have shifted their attention to document-level RE that leverages the whole input document to extract relational triples.

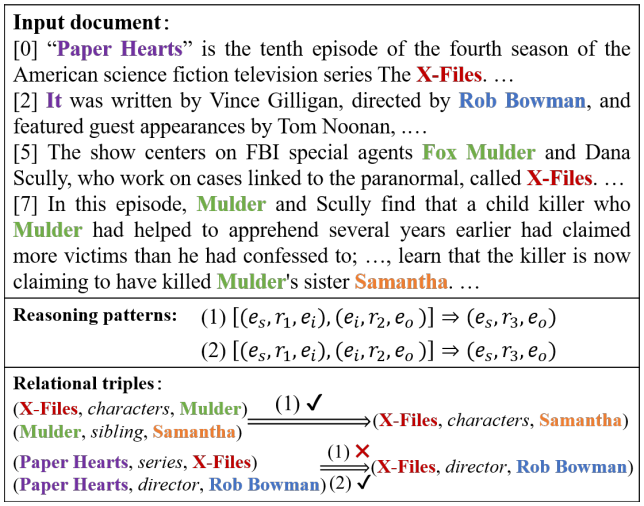

Figure 1: An example with two reasoning patterns from the DocRED dataset. e_i denotes an arbitrary entity in the document. In the box at the bottom, the left relational triples are clearly expressed in the document, while the right ones are not explicitly expressed but can be inferred via different reasoning patterns. The arrow in the middle indicates the reasoning patterns used to infer the right relational triple.

However, one of the primary challenges of document-level RE is to predict *implicit relations* between entities. Usually, these relations are not explicitly expressed in the

* Corresponding Author.

document and can be extracted via *relational reasoning*, which aims to exploit the dependence among entities and entity pairs to infer implicit relations. For example, in Figure 1, the relation *characters* between *X-Files* and *Samantha* is an implicit one, whose prediction can be refined by exploiting the information of the other two entity pairs, (*X-Files, Mulder*) and (*Mulder, Samantha*). To effectively model relational reasoning, most existing methods (Guo et al. 2019; Zeng et al. 2020; Nan et al. 2020; Xu et al. 2021) utilize graph neural networks (GNNs) to capture the entity- or mention-level dependence. Meanwhile, since Transformer (Vaswani et al. 2017) can effectively model long-range dependence, some studies (Tang et al. 2020; Zhou et al. 2021) directly utilize pre-trained language models (PLMs) to learn better entity or entity pair representations for implicit relation predictions. Furthermore, some recent studies (Zhang et al. 2021; Tan et al. 2022) focus on leveraging the dependence among entity pairs to infer implicit relations between entities. In spite of their success, they still have inherent defects. **First**, they usually only consider the first reasoning pattern shown in the middle box of Figure 1, which, however, only covers limited relational triples. Back to the bottom box of Figure 1, the relation between *X-Files* and *Rob Bowman* can only be successfully predicted via the second reasoning pattern. **Second**, they do not explicitly model the process of relational reasoning during training, which is unable to fully exert the potential of relational reasoning.

To deal with the first problem, in this paper, we propose a document-level RE model with a reasoning module that can effectively exploit the dependence among entity pairs for relational reasoning. As shown in Figure 2(b), reasoning module contains a core unit, the *reasoning multi-head self-attention* (R-MSA) unit, which is a variant of the conventional multi-head self-attention and utilizes four attention heads to model four common reasoning patterns (See Table 1), respectively. Compared with previous studies (Zeng et al. 2020; Zhang et al. 2021; Tan et al. 2022) that mainly consider the first pattern, our reasoning patterns are more comprehensive and have higher coverage of relational triples.

To address the second issue, we propose a *self-distillation* training framework, which can significantly enhance the reasoning ability of our model. As shown in Figure 3, our framework consists of two branches. **In the first branch**, we randomly mask some entity pair feature vectors in the document and treat their relations as pseudo implicit ones. Then, we train our reasoning module to infer these relations by exploiting the feature information of other related entity pairs. By doing so, we can explicitly model the process of relational reasoning, which can provide our model with more explicit reasoning supervision signals. However, because this masking operation is not used during testing, it causes an input gap between training and testing scenarios, which would hurt the model performance. To bridge this gap, **in the second branch**, we conduct conventional supervised training without masking operation, which is consistent with the testing scenario of our model. Meanwhile, we use a Kullback-Leibler (KL) divergence loss to minimize the difference between the predictions of the two branches. In this way, we can transfer the reasoning ability learned from the training of the first branch to the testing scenario. In particular, to further improve our model, we employ a curriculum learning strategy to dynamically select masked entity pairs in an easy-to-hard manner. To demonstrate the effectiveness and generality of our model, we conduct comprehensive experiments on three public datasets, of which results show that our model consistently outperforms all competitive baselines.

Our Model

In this section, we describe the proposed model in detail. As illustrated in Figure 2(a), our model consists of two components: an encoder and a reasoning module. We detail our encoder and reasoning module in Section and Section , respectively, and then introduce a novel self-distillation training framework for our model in Section .

Encoder

We employ a pre-trained language model (PLM) as our encoder to learn better contextual representations of entities. Following Zhang et al. (2021), we then use these representations to construct an entity pair feature matrix, so as to facilitate the computation of reasoning module.

Formally, an input document D often contains multiple entities $\{e_i\}_{i=1}^{N}$, where each entity e_i may occur multiple times as mentions $\{m_j^i\}_{j=1}^{N_{e_i}}$. Following Zhou et al. (2021), we first mark the position of each mention in the input document by inserting a special symbol "$*$" at its start and end positions. Then, we feed the document into the PLM to obtain its contextual embeddings, $\mathbf{H} = [h_1, h_2, ..., h_{|D|}]$. Here, we take the contextual embedding of the special token "$*$" at the start of each mention as its embedding, and then employ *logsumexp pooling* (Jia, Wong, and Poon 2019) to obtain the representation $h(e_i)$ of entity e_i by aggregating all its mention embeddings: $h(e_i) = \log \sum_{j=1}^{N_{e_i}} \exp(h(m_j^i))$.

Then, the feature vector $F_{s,o}$ of the entity pair (e_s,e_o) is calculated via a feed-forward neural network (FNN):

$$F_{s,o} = \mathrm{FNN}\big(\big[\tanh(W_s[h(e_s); c_{s,o}]); \tanh(W_o[h(e_o); c_{s,o}])\big)\big], \tag{1}$$

where W_o and W_s are learnable weight matrices, and $c_{s,o}$ denotes the localized context embedding (Zhou et al. 2021) encoding the contextual information specific to (e_s,e_o). More specifically, $c_{s,o}$ is computed as

$$c_{s,o} = \mathbf{H}^{\mathsf{T}} \frac{A_s \circ A_o}{\mathbf{1}^{\mathsf{T}}(A_s \circ A_o)}, \tag{2}$$

where A_s and A_o denote the PLM last-layer attention weights of entities e_s and e_o to all tokens in the document, respectively, and $\circ$ refers to element-wise multiplication.

Finally, all entity pair feature vectors within the document are merged to form an entity pair feature matrix $M^{(0)}$=$[F_{s,o}]_{N\times N}$, where each row $M_{s,*}^{(0)}$ corresponds to a subject entity e_s and each column $M_{*,o}^{(0)}$ corresponds to an object entity e_o.

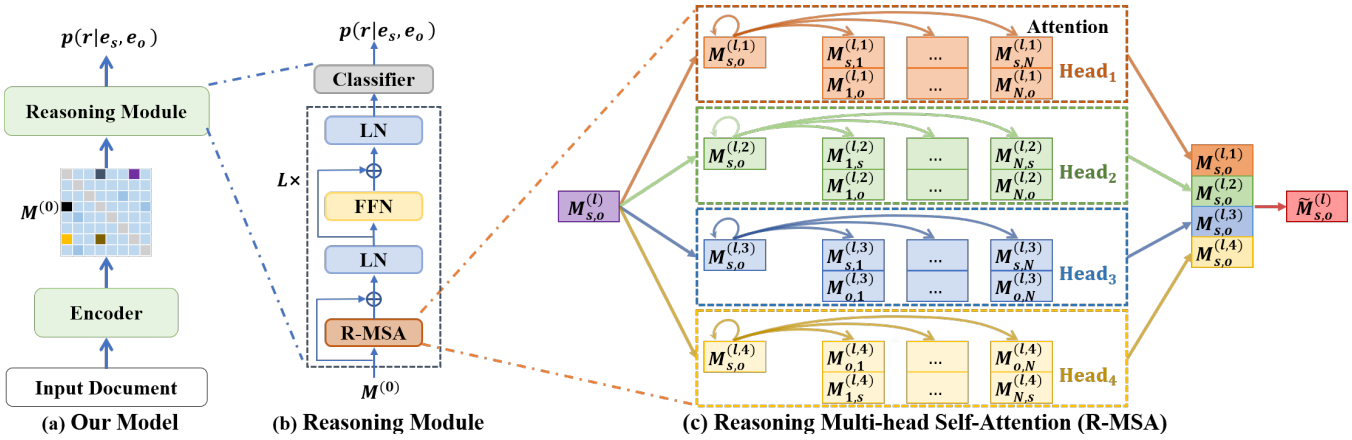

Figure 2: Our model is composed of an encoder and a reasoning module, where reasoning module consists of L reasoning layers and a classifier. Each reasoning layer contains a core component, the R-MSA unit, which is a variant of the conventional multi-head self-attention and utilizes four attention heads to model four common reasoning patterns, respectively.

Reasoning Pattern	Example	Rate
(1) $[(e_s, r_1, e_i), (e_i, r_2, e_o)] \Rightarrow (e_s, r_3, e_o)$	[(Bob, father, Danny), (Danny, spouse, Anna)]⇒(Bob, mother, Anna)	24.83%
(2) $[(e_i, r_1, e_s), (e_i, r_2, e_o)] \Rightarrow (e_s, r_3, e_o)$	[(Bob, brother, Harry), (Bob, father, Danny)]⇒(Harry, father, Danny)	19.28%
(3) $[(e_s, r_1, e_i), (e_o, r_2, e_i)] \Rightarrow (e_s, r_3, e_o)$	[(Bob, father, Danny), (Harry, father, Danny)]⇒(Bob, brother, Harry)	24.69%
(4) $[(e_o, r_1, e_i), (e_i, r_2, e_s)] \Rightarrow (e_s, r_3, e_o)$	[(Bob, mother, Anna), (Anna, spouse, Danny)]⇒(Danny, child, Bob)	7.70%

Table 1: Illustration of the four common reasoning patterns. ⇒ denotes the reasoning operation. In the last column, we calculate the ratios of relational triples that can be inferred by these reasoning patterns on the DocRED dataset. Please note that unlike previous methods that generally only consider the first pattern, our model models these four patterns, as shown in Figure 2(c).

Reasoning Module

Based on the entity pair feature matrix $M^{(0)}$, our reasoning module aims to learn more expressive entity pair representations, with which we can infer implicit relations between entities. As shown in Figure 2(b), the reasoning module is composed of L reasoning layers, stacked by a classifier. Each reasoning layer contains four components: a *reasoning multi-head self-attention* (R-MSA) unit, a FFN unit, and two layer normalization sublayers. Back to Figure 2(c), the R-MSA unit is a variant of the conventional multi-head self-attention, which is equipped with four attention heads to model four common reasoning patterns, respectively. Table 1 illustrates these four reasoning patterns and their corresponding examples. We find that these four patterns can cover significantly more relational triples than previous studies (Zeng et al. 2020; Zhang et al. 2021; Tan et al. 2022) that generally only consider the first pattern.

Since the calculation procedures of all R-MSA heads are similar, we take the first head as an example to elaborate its details. Specifically, for the entity pair (e_s,e_o), at the $(l+1)$-th reasoning layer, we first concatenate the corresponding entity pair feature vectors in the s-th row and o-th column of the entity pair feature matrix $M^{(l)}$, and then reduce their dimensions through a single linear layer:

$$F_i^{(l,1)} = W_d\left[M_{s,i}^{(l)}; M_{i,o}^{(l)}\right] + b_d, \quad i = \{1, 2, \cdots, N\}, \quad (3)$$

where W_d and b_d are trainable parameters, and $[\cdot\,;\cdot]$ represents the concatenation operation. Next, we obtain the output vector $M_{s,o}^{(l,1)}$ of the first R-MSA head for entity pair (e_s,e_o) through an attention mechanism:

$$\begin{aligned} &M_{s,o}^{(l,1)} = \text{Attention}\left(Q, K, V\right), \\ &\text{where } \; Q = M_{s,o}^{(l)}, \; K = V = \left[M_{s,o}^{(l)}; F_1^{(l,1)}; \cdots; F_N^{(l,1)}\right]. \end{aligned} \quad (4)$$

Note that the two superscripts of $M_{s,o}^{(l,1)}$ and $F_i^{(l,1)}$ indicate the reasoning layer index and attention head index, respectively, and their subscripts are the entity indexes. Afterwards, we aggregate the outputs of all R-MSA heads to produce the output of the R-MSA unit:

$$\widetilde{M}_{s,o}^{(l)} = \text{LN}\left(M^{(l)} + W_O\left[M_{s,o}^{(l,1)}; \cdots; M_{s,o}^{(l,4)}\right] + b_O\right), \quad (5)$$

where W_O and b_O are model parameters, and $\text{LN}(\cdot)$ is the layer normalization (Ba, Kiros, and Hinton 2016) function.

Finally, the output of the $(l+1)$-th reasoning layer is computed as follows:

$$M^{(l+1)} = \text{LN}\left(\widetilde{M}^{(l)} + \text{FNN}(\widetilde{M}^{(l)})\right), \quad (6)$$

where $\widetilde{M}^{(l)} = [\widetilde{M}_{s,o}^{(l)}]_{N\times N}$. After repeating the above process L times, we can get a more expressive feature matrix $M^{(L)}$.

Based on $M^{(L)}$, we use a single-layer classifier to predict the relational probability distribution for entity pair (e_s,e_o):

$$p(r|e_s, e_o) = \sigma(W_c M_{s,o}^{(L)} + b_c), \quad (7)$$

where W_c and b_c are learnable parameters of the classifier.

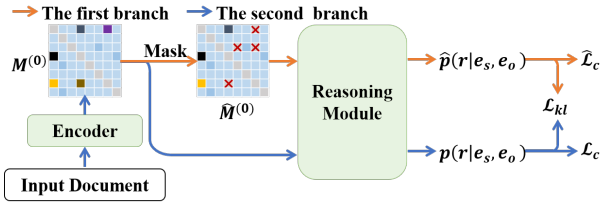

Figure 3: Illustration of our self-distillation training framework, which consists of two branches. In the first branch, we explicitly model the process of relational reasoning. In the second branch, we do not perform masking operation, which is consistent with the testing scenario of our model.

Model Training

To better train our reasoning module, we propose a self-distillation training framework. As shown in Figure 3, our framework consists of two branches.

The first branch. Inspired by Masked Language Modeling (MLM) (Devlin et al. 2019), in this branch, we first randomly select some entity pairs and replace their feature vectors in $M^{(0)}$ with a special [Mask] vector, forming a new feature matrix $\widehat{M}^{(0)}$. Then, we feed $\widehat{M}^{(0)}$ into reasoning module to predict the relation probability distributions of all entity pairs, denoted by $\{\widehat{p}(r|e_s,e_o)\}$. Finally, we train reasoning module to accurately infer the relations of the masked entity pairs. By doing so, we explicitly model the process of relational reasoning, which can provide reasoning module with more explicit reasoning supervisory signals. Note that during testing, we do no mask any entity pair feature vector because it may cause the model performance degrade. Thus, this masking operation leads to the input gap between the training and testing of our model.

The second branch. To reduce the above gap, in this branch, we directly input the original entity pair feature matrix M^0 to reasoning module, obtaining the relation probability distributions $\{p(r|e_s,e_o)\}$. Note that this branch does not perform the above masking operation and thus is compatible with the testing scenario. Afterwards, we introduce a KL divergence loss $\mathcal{L}_{kl}$ to minimize the difference between $p(r|e_s,e_o)$ and $\widehat{p}(r|e_s,e_o)$, formulated as

$$\mathcal{L}_{kl} = \mathrm{KL}\big(p(r|e_s,e_o)||\widehat{p}(r|e_s,e_o)\big). \tag{8}$$

Training objective. Using ground-truth labels as supervisory signals, we also introduce two classification losses, $\mathcal{L}_c$ and $\widehat{\mathcal{L}}_c$, to supervise the relation predictions of the two branches. Thus, the final training objective of our model is formulated as

$$\mathcal{L} = \mathcal{L}_c + \widehat{\mathcal{L}}_c + \mathcal{L}_{kl}. \tag{9}$$

To address the multi-label and imbalanced label distribution problems, we adapt adaptive thresholding loss (Zhou et al. 2021) to model our classification losses $\mathcal{L}_c$ and $\widehat{\mathcal{L}}_c$. Specifically, we introduce a special relation class TH and use its logits $\mathrm{logit}_{\mathrm{TH}}$ as the adaptive threshold value for each entity pair, where we expect that all logits of target relation classes $\mathcal{R}_{pos}$ are greater than $\mathrm{logit}_{\mathrm{TH}}$ while all logits of non-target relation classes $\mathcal{R}_{neg}$ are less than $\mathrm{logit}_{\mathrm{TH}}$:

$$\mathcal{L}_c = -\left(\sum_{r\in\mathcal{R}_{pos}} \log\left(\frac{\exp(\mathrm{logit}_r)}{\sum_{r'\in\{\mathcal{R}_{pos},\mathrm{TH}\}}\exp(\mathrm{logit}_{r'})}\right)\right) - \log\left(\frac{\exp(\mathrm{logit}_{\mathrm{TH}})}{\sum_{r'\in\{\mathcal{R}_{neg},\mathrm{TH}\}}\exp(\mathrm{logit}_{r'})}\right). \tag{10}$$

Curriculum Learning. To further improve the training of our model, in the first branch, we employ a curriculum learning strategy to dynamically select masked entity pairs in an easy-to-hard manner. We uniformly sample γ_t percent of entity pairs from the entity pair feature matrix M^0 as the masked ones. Intuitively, a greater mask rate may make model training more difficult. Therefore, to better train our model, we begin training with a small mask rate and linearly increase it to the maximum mask rate γ_{max}: $\gamma_t = \min(\gamma_{max}, \frac{t}{T})$, where t is the current training step, and T is the maximal training step.

Experiments

Datasets

We evaluate our model on three commonly-used datasets:

- **DocRED** (Yao et al. 2019). It is a large-scale human-annotated dataset for document-level RE, which is constructed from Wikipedia and Wikidata. It contains 96 target relations, 132,275 entities, and 56,354 relationship triples in total. In DocRED, more than 40.7% of relational facts can only be extracted from multiple sentences, and 61.1% of relational triples require relational reasoning. We follow the standard split of the dataset, 3,053 documents for training, 1,000 for development and, 1,000 for the test.
- **CDR** (Li et al. 2016). It is a biomedical dataset and consists of 1,500 PubMed abstracts, which are equally divided into three sets for training, development, and testing. On this dataset, the model is expected to predict the binary relations between Chemical and Disease entities.
- **GDA** (Wu et al. 2019). This dataset is a large-scale biomedical one, which is constructed from MEDLINE abstracts by method of distant supervision. GDA contains 29,192 documents as the training set and 1,000 as the test set. It contains only one target relation between Chemical and Disease entities, i.e., *Chemical-Induced-Disease*. We follow Tang et al. (2020) to divide the training set into two parts, 23,353 documents for training and 5,839 for development.

Settings

Using PyTorch, we develop our model based on Huggingface's Transformers (Wolf et al. 2020). We use BERT-base (Devlin et al. 2019) or RoBERTa-large (Liu et al. 2019) as the encoder on DocRED, and SciBERT-base (Beltagy, Lo, and Cohan 2019) on CDR and GDA. We employ AdamW (Loshchilov and Hutter 2019) to optimize our model with a linear warmup (Goyal et al. 2017) for the first 6% steps. We empirically set the layer number L of reasoning module to 2. We apply dropout (Srivastava et al. 2014) between layers

Model	Dev					Test	
	IgnF_1	F_1	Intra-F_1	Inter-F_1	Infer-Ac	IgnF_1	F_1
GEDA-BERT (Li et al. 2020)†	54.52	56.16	–	–	–	53.71	55.74
LSR-BERT (Nan et al. 2020)†	52.43	59.00	65.26	52.05	–	56.97	59.05
GLRE-BERT (Wang et al. 2020)†	–	–	–	–	–	55.40	57.40
GAIN-BERT (Zeng et al. 2020)†	59.14	61.22	67.10	53.90	58.42*	59.00	61.24
HeterGSAN-BERT (Xu et al. 2021)†	58.13	60.18	–	–	–	57.12	59.45
SSAN-BERT (Xu et al. 2021)†	56.68	58.95	–	–	–	56.06	58.41
BERT-base (Wang et al. 2019)†	–	54.16	61.61	47.15	–	–	53.20
BERT-TS (Wang et al. 2019)†	–	54.42	61.80	47.28	–	–	53.92
HIN-BERT (Tang et al. 2020)†	54.29	56.31	–	–	–	53.70	55.60
CorefBERT (Ye et al. 2020)†	55.32	57.51	–	–	–	54.54	56.96
ATLOP-BERT (Zhou et al. 2021)†	59.22	61.09	–	–	58.29*	59.31	61.30
DocuNet-BERT (Zhang et al. 2021)†	59.86	61.83	–	–	–	59.93	61.86
SIRE-BERT (Zeng et al. 2021)†	59.82	61.60	68.07	54.01	–	60.18	62.05
KD-BERT (Tan et al. 2022)†	60.08	62.03	–	–	58.93*	60.04	62.08
Ours-BERT(SD→KD)	59.83	61.76	68.12	54.09	59.31	59.94	61.81
Ours-BERT(SD→R-Drop)	60.12	61.92	68.39	54.92	59.74	60.11	62.03
Ours-BERT	**60.85±0.10**	**62.81±0.13**	**68.67±0.11**	**56.09±0.21**	**61.08±0.18**	**60.91**	**62.85**

Table 2: Experimental results on the development and test sets of DocRED. We report the mean and standard deviation on the development set by conducting five experiments with different random seeds. Besides, we report the official test scores of the best checkpoint on the development set. † indicates original paper scores. Results with * are obtained by our reproduction. KD denotes the vanilla knowledge distillation and SD means our self-distillation training framework. SD→KD (SD→R-Drop) means to replace our SD with KD (R-Drop).

with rate 0.1, and clip the gradients of model parameters to a maximal norm of 1.0. All hyper-parameters are tuned on the development set.

Baselines

We compare our model with the following baselines:

GNN-based Models. These models build task-specific document graphs to capture the dependence among entities or mentions for relational reasoning. Here, we consider EoG (Christopoulou et al. 2019), DHG (Zhang et al. 2020), GEDA (Li et al. 2020), LSR (Nan et al. 2020), GLRE (Wang et al. 2020), GAIN (Zeng et al. 2020), HeterGSAN (Xu et al. 2021), and SSAN (Xu et al. 2021) for comparison.

Transformer-based Models. These models aim to extract more useful information from PLMs to enhance entity or entity pair representations for better relation predictions. Our considered baselines include BERT-base (Wang et al. 2019), BERT-TS (Wang et al. 2019), HIN-BERT (Tang et al. 2020), CorefBERT (Ye et al. 2020), and ATLOP-BERT (Zhou et al. 2021).

Furthermore, we also compare our model with some recent studies, including DocuNet (Zhang et al. 2021), SIRE (Zeng et al. 2021), and KD (Tan et al. 2022). Similar to us, they also attempt to exploit the dependence among entity pairs for relational reasoning.

Effect of Maximum Mask Rate γ_{max}

We first investigate the effect of the hyper-parameter γ_{max} on our model. To this end, we conduct an experiment with different γ_{max} on the DocRED dataset, of which results are shown in Figure 4. We observe that our model is not very sensitive to γ_{max}, and achieves the best performance when γ_{max} is set to 0.2. Therefore, we set γ_{max}=0.2 for efficiency in subsequent experiments.

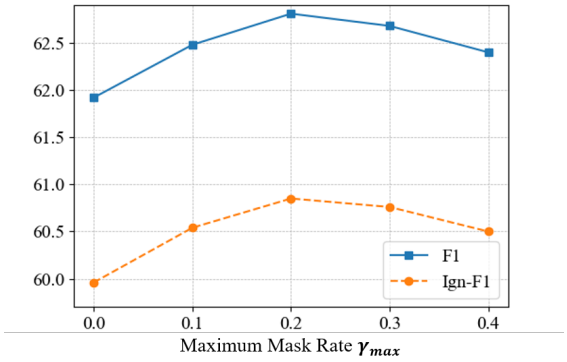

Figure 4: The performance of our model with different maximum mask rates γ_{max} on the development set of DocRED.

Results on the DocRED Dataset

Following Yao et al. (2019), we report F_1 and IgnF_1 scores on the DocRED dataset, where IgnF_1 is computed by excluding relational triplets appearing in the training set. From Table 2, we can find that our model consistently outperforms all competitive baselines. Moreover, we draw several interesting conclusions:

First, compared with our base model, ATLOP-BERT, our model achieves improvements of **1.55** F_1 and **1.60** IgnF_1 points on the test set. Note that ATLOP-BERT is essentially a variant of our model, where our reasoning module and self-distillation training framework are removed. Thus, these

Model	CDR	GDA
BRAN (Verga et al. 2018)	62.1	–
EoG (Christopoulou et al. 2019)	63.6	81.5
LSR (Nan et al. 2020)	64.8	82.2
DHG (Zhang et al. 2020)	65.9	83.1
GLRE (Wang et al. 2020)	68.5	–
SciBERT (Beltagy, Lo, and Cohan 2019)	65.1	82.5
ATLOP-SciBERT (Zhou et al. 2021)	69.4	83.9
DocuNet-SciBERT(Zhang et al. 2021)	76.3	85.3
Ours-SciBERT	**76.8**	**86.4**

Table 3: The F_1 scores on the CDR and GDA test sets.

results clearly demonstrate that our reasoning module and training framework can effectively improve document-level RE. Besides, we note that GNN-based models achieve worse results than models that exploit the dependence among entity pairs for relational reasoning, such as DocuNet-BERT and KD-BERT. This observation suggests that the dependence among entity pairs is more useful for relational reasoning than the dependence among entities or mentions.

Second, we follow Nan et al. (2020) to also report Intra-F_1/Inter-F_1 scores in Table 2, where these two metrics only consider intra- and inter-sentence relations, respectively. Although our model achieves a slight gain on Intra-F_1, it yields a significant improvement of **2.08** Inter-F_1 points over the competitive baseline SIRE-BERT. This reveals that the advantage of our model lies in extracting cross-sentence relations, most of which require the help of relational reasoning.

Third, in the bottom box of Table 2, we also train our model utilizing the vanilla knowledge distillation (Hinton et al. 2015) and R-Drop (Wu et al. 2021a), respectively. Compared with these two variants, our model still achieves better performance, indicating that our self-distillation training framework can stimulate the reasoning ability of our model more effectively.

Fourth, we report a new metric, Infer-Ac, to verify that our model can effectively model four common reasoning patterns (See Table 1). Unlike previous metrics, Infer-Ac measures the prediction accuracy of relation triples that conform to this four reasoning patterns in the dataset. On this metric, our model also significantly exceeds previous baselines.

Results on the Biomedical Datasets

As shown in Table 3, our model still outperforms all baselines on two biomedical datasets, GDA and CDR, demonstrating that our model is also general to the biomedical domain. Moreover, we find that the improvement of our model on GDA is more significant than that on CDR. The underlying reason is that our self-distillation training framework is more advantageous in the scenario of large amounts of data.

Ablation Study

Then, we remove different components from our model, and show the performance of our model variants in Table 4.

(1) *w/ R-MSA→MSA*. In this variant, we replace our R-MSA unit with the standard multi-head self-attention (MSA) that directly uses the information of all other entity pairs to enhance the representation of each considered one. As shown in Line 3 of Table 4, this replacement causes a significant performance drop. For this result, we speculate that MSA introduces much noise, and thus it cannot effectively infer the relations of masked entity pairs during training. Besides, since this variant has the same number of parameters as our model, this result also proves that the performance gain of our model does not stem from the increase of parameters.

Model	IgnF_1	F_1
Ours-BERT	**60.85**	**62.81**
w/ R-MSA→MSA	57.45	59.39
w/ Only the first reasoning pattern	60.25	62.16
w/o The first branch	59.58	61.53
w/o The second branch	60.46	62.38
w/o Curriculum Learning	60.61	62.56

Table 4: Ablation study of our model on the development set of DocRED.

(2) *w/ Only the first reasoning pattern*. Similar to previous studies that only consider the first reasoning pattern (e.g., SIRE-BERT and KD-BERT in Table 2), we only model the first reasoning pattern in this variant. Back to Line 4 of Table 4, we also note that this variant is inferior to our model, indicating that our four reasoning patterns indeed cover more relational triplets. Furthermore, this variant still outperforms the above baselines, indirectly proving the effectiveness of our self-distillation training framework.

(3) *w/o The first branch*. When we remove the first branch from our self-distillation training framework, the performance of our model sharply drops (See Line 5 of Table 4). Thus, we confirm that the training of the first branch can effectively enhance the reasoning ability of our model.

(4) *w/o The second branch*. We train this variant only through the first branch, which, however, leads to a performance decline (See Line 6 of Table 4). The underlying reason is that the masking operation in the first branch causes the input gap between training and testing of our model.

(5) *w/o Curriculum Learning*. In this variant, we discard our curriculum learning strategy. As shown in Line 7 of Table 4, the performance of our model drops, proving that this strategy is important for the training of our model.

Reasoning Performance

Following Zeng et al. (2020), we compare models in terms of Infer-F_1, aiming to evaluate the multi-hop reasoning ability of models. From Table 5, we observe that our model significantly outperforms GAIN-BERT by **3.22** Infer-F_1 points. Meanwhile, removing the first branch or reasoning module from our model results in a significant performance drop. These results show that both reasoning module and the self-distillation training framework can effectively improve the reasoning ability of our model.

Furthermore, we conduct an interesting experiment to further verify the reasoning ability and robustness of our model. Specifically, we also perform masking operation with different mask rates during testing, of which experimental results

Model	Infer-F_1	P	R
GAIN-GloVe	40.82	32.76	54.14
SIRE-GloVe	42.72	34.83	55.22
BERT-RE	39.62	34.12	47.23
GAIN-BERT	46.89	38.71	59.45
Ours-BERT	**50.11**	**42.99**	**60.05**
w/o The first branch	47.92	40.03	59.68
w/o Reasoning module	46.62	38.42	59.29

Table 5: Infer-F_1 scores on the development set of DocRED.

are reported in Figure 5. When the mask rate is less than 0.3 during testing, the performance of our model only drops slightly. This is because our model is trained with a mask rate of 0.2. Surprisingly, even if the mask rate increases to 0.8, our model still achieves an F_1 score of 55.17, outperforming that of BERT-TS (Wang et al. 2019) in Table 2.

Related Work

Sentence-level RE has received considerable attention in the past decade. Many approaches have been proposed to tackle this task effectively (Zeng et al. 2015; Wang et al. 2016; Zhang et al. 2017; Feng et al. 2018; Yu et al. 2020; Wu et al. 2021b). However, a large number of relational facts are expressed across sentences in real-world applications (Verga, Strubell, and McCallum 2018; Yao et al. 2019). Therefore, some recent studies have shifted their attention to document-level RE (Zeng et al. 2020; Zhou et al. 2021; Zhang et al. 2021; Tan et al. 2022).

In this regard, due to the advantages of GNNs in relational reasoning, many GNN-based models have been proposed for document-level RE (Guo et al. 2019; Zeng et al. 2020; Nan et al. 2020; Xu et al. 2021). Generally, they first construct a document graph utilizing dependency structures, heuristic rules, or structured attention, and then employ GNNs (Liang et al. 2016; Guo et al. 2019) to perform reasoning on this graph. Nan et al. (2020) builds a latent document-level graph based on the structured attention mechanism and proposes a refinement strategy to enable the model to incrementally aggregate relevant information for multi-hop reasoning. Zeng et al. (2020) constructs two graphs, mention-level and entity-level graphs, to model the dependence among entities and mentions, respectively. Xu et al. (2021) introduces a path reconstructor into the document graph, which ensures the model pays more attention to entity pairs with relations.

Meanwhile, considering the transformer architecture can implicitly model long-distance dependencies, some studies directly utilize PLMs to learn better entity or entity pair representations for document-level RE (Wang et al. 2019; Zhou et al. 2021; Tang et al. 2020; Zhang et al. 2022). For example, Zhou et al. (2021) proposes adaptive thresholding loss and localized context pooling to solve the multi-label and multi-entity problems. Furthermore, some researchers put efforts into leveraging the dependence among entity pairs to infer implicit relations between entities (Zhang et al. 2021; Tan et al. 2022). Zhang et al. (2021) reformulates document-level RE task as a semantic segmentation problem and uses

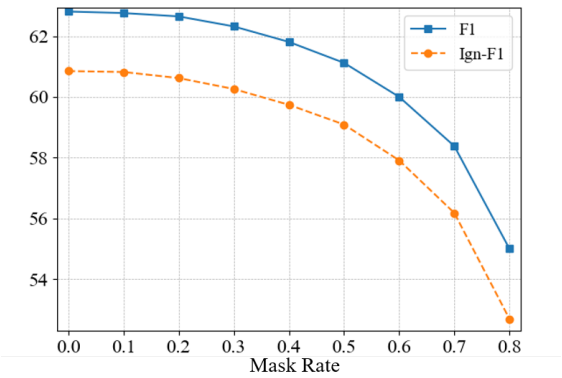

Figure 5: The performance of our model with different mask rates during testing on the development set of DocRED.

CNNs to capture the dependence among entity pairs. However, the above approaches suffer from two obvious drawbacks: 1) they often consider only one reasoning pattern (See the first reasoning pattern in Table 1), of which coverage on relational triple is limited; 2) they do not explicitly model the process of relational reasoning. These two defects seriously limit the performance of document-level RE models.

In this work, we propose a document-level RE model with a reasoning module that comprehensively considers four common reasoning patterns. Besides, inspired by recent studies of self-distillation methods in the communities of CV and NLP (Clark et al. 2019; Zeng et al. 2019; Zhang et al. 2019; Liu et al. 2020; Wu et al. 2022; Kong et al. 2022; Zhou et al. 2022), we explore a self-distillation based relational reasoning training framework for document-level RE, which explicitly models the process of relational reasoning. To the best of our knowledge, our work is the first attempt to explore the self-distillation framework to enhance the relational reasoning ability of the document-level RE model. Finally, please note that our self-distillation training framework is also related to R-Drop (Wu et al. 2021a). However, the difference between them is that we perform masking operation on the first branch to train the reasoning ability of the model more effectively, and the experimental results also prove that our framework is superior to R-Drop.

Conclusion and Future Work

In this paper, we have proposed a document-level RE model, which simultaneously models four common reasoning patterns to better infer the implicit relations between entities. Furthermore, we have proposed a self-distillation training framework for our model. By explicitly modeling the relational reasoning process, this framework is able to provide more explicit supervisory signals for the relational reasoning training of our model. Experimental results on three commonly-used datasets demonstrate that our model outperforms all existing competitive baselines.

In further, we will extend our training framework to a semi-supervised setting. Besides, we plan to apply our model to other relational reasoning tasks, such as Knowledge Graph Completion, so as to verify its generality.

Acknowledgments

The authors would like to thank the three anonymous reviewers for their comments on this paper. This research was supported in part by the National Natural Science Foundation of China under Grant Nos. 62076211, U1908216, 62276219, and 61573294.

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. In *arXiv preprint arXiv:1607.06450*.

Baldini Soares, L.; FitzGerald, N.; Ling, J.; and Kwiatkowski, T. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *ACL 2019*.

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *EMNLP 2019*.

Christopoulou, F.; Miwa, M.; Ananiadou, S.; and Ananiadou, S. 2019. Connecting the Dots: Document-level Neural Relation Extraction with Edge-oriented Graphs. In *EMNLP 2019*.

Clark, K.; Luong, M.-T.; Khandelwal, U.; Manning, C. D.; and Le, Q. 2019. BAM! Born-Again Multi-Task Networks for Natural Language Understanding. In *ACL 2019*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL 2019*.

Dong, L.; Wei, F.; Zhou, M.; and Xu, K. 2015. Question answering over freebase with multi-column convolutional neural networks. In *ACL 2015*.

Feng, J.; Huang, M.; Zhao, L.; Yang, Y.; and Zhu, X. 2018. Reinforcement Learning for Relation Classification From Noisy Data. In *AAAI 2018*.

Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; and He, K. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. In *arXiv preprint arXiv:1706.02677*.

Guo, Z.; Zhang, Y.; Lu, W.; Guo, Z.; Zhang, Y.; and Lu, W. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *ACL 2019*.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. In *arXiv preprint arXiv:1503.02531*.

Jia, R.; Wong, C.; and Poon, H. 2019. Document-Level N-ary Relation Extraction with Multiscale Representation Learning. In *NAACL 2019*.

Kong, X.; Gao, Z.; Li, X.; Hong, M.; Liu, J.; Wang, C.; Xie, Y.; and Qu, Y. 2022. En-Compactness: Self-Distillation Embedding & Contrastive Generation for Generalized Zero-Shot Learning. In *CVPR 2022*.

Li, B.; Ye, W.; Sheng, Z.; Xie, R.; Xi, X.; and Zhang, S. 2020. Graph Enhanced Dual Attention Network for Document-Level Relation Extraction. In *COLING 2020*.

Li, J.; Sun, Y.; Johnson, R. J.; Sciaky, D.; Wei, C.-H.; Leaman, R.; Davis, A. P.; Mattingly, C. J.; Wiegers, T. C.; and Lu, Z. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. In *Database*.

Liang, X.; Shen, X.; Feng, J.; Lin, L.; and Yan, S. 2016. Semantic object parsing with graph lstm. In *ECCV 2016*.

Liu, X.; Liu, K.; Li, X.; Su, J.; Ge, Y.; Wang, B.; and Luo, J. 2020. An Iterative Multi-Source Mutual Knowledge Transfer Framework for Machine Reading Comprehension. In *IJCAI 2020*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. In *arXiv preprint arXiv:1907.11692*.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR 2019*.

Nan, G.; Guo, Z.; Sekulic, I.; and Lu, W. 2020. Reasoning with Latent Structure Refinement for Document-Level Relation Extraction. In *ACL 2020*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR 2014*.

Tan, Q.; He, R.; Bing, L.; and Ng, H. T. 2022. Document-Level Relation Extraction with Adaptive Focal Loss and Knowledge Distillation. In *ACL Findings 2022*.

Tang, H.; Cao, Y.; Zhang, Z.; Cao, J.; Fang, F.; Wang, S.; and Yin, P. 2020. Hin: Hierarchical inference network for document-level relation extraction. In *PAKDD 2020*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NeurIPS 2017*.

Verga, P.; Strubell, E.; and McCallum, A. 2018. Simultaneously Self-Attending to All Mentions for Full-Abstract Biological Relation Extraction. In *NAACL 2018*.

Verga, P.; Strubell, E.; McCallum, A.; Strubell, E.; and McCallum, A. 2018. Simultaneously Self-Attending to All Mentions for Full-Abstract Biological Relation Extraction. In *NAACL 2018*.

Wang, D.; Hu, W.; Cao, E.; and Sun, W. 2020. Global-to-Local Neural Networks for Document-Level Relation Extraction. In *EMNLP 2020*.

Wang, H.; Focke, C.; Sylvester, R.; Mishra, N.; and Wang, W. 2019. Fine-tune bert for docred with two-step process. In *arXiv preprint arXiv:1909.11898*.

Wang, L.; Cao, Z.; de Melo, G.; and Liu, Z. 2016. Relation Classification via Multi-Level Attention CNNs. In *ACL 2016*.

Wang, Q.; Mao, Z.; Wang, B.; and Guo, L. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Le Scao, T.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. 2020. Transformers: State-of-the-Art Natural Language Processing. In *EMNLP 2020*.

Wu, C.; Cao, L.; Ge, Y.; Liu, Y.; Zhang, M.; and Su, J. 2022. A Label Dependence-aware Sequence Generation Model for Multi-level Implicit Discourse Relation Recognition. In *AAAI 2022*.

Wu, L.; Li, J.; Wang, Y.; Meng, Q.; Qin, T.; Chen, W.; Zhang, M.; Liu, T.-Y.; et al. 2021a. R-drop: Regularized dropout for neural networks. In *NeurIPS 2021*.

Wu, T.; Li, X.; Li, Y.-F.; Haffari, R.; Qi, G.; Zhu, Y.; and Xu, G. 2021b. Curriculum-meta learning for order-robust continual relation extraction. In *AAAI 2021*.

Wu, Y.; Luo, R.; Leung, H. C.; Ting, H.-F.; and Lam, T.-W. 2019. Renet: A deep learning approach for extracting gene-disease associations from literature. In *RECOMB 2019*.

Xu, W.; Chen, K.; Zhao, T.; and Zhao, T. 2021. Document-level relation extraction with reconstruction. In *AAAI 2021*.

Yao, Y.; Ye, D.; Li, P.; Han, X.; Lin, Y.; Liu, Z.; Liu, Z.; Huang, L.; Zhou, J.; and Sun, M. 2019. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In *ACL 2019*.

Ye, D.; Lin, Y.; Du, J.; Liu, Z.; Li, P.; Sun, M.; and Liu, Z. 2020. Coreferential Reasoning Learning for Language Representation. In *EMNLP 2020*.

Yu, H.; Zhang, N.; Deng, S.; Ye, H.; Zhang, W.; and Chen, H. 2020. Bridging Text and Knowledge with Multi-Prototype Embedding for Few-Shot Relational Triple Extraction. In *COLING 2020*.

Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In *ACL 2015*.

Zeng, J.; Liu, Y.; Su, J.; Ge, Y.; Lu, Y.; Yin, Y.; and Luo, J. 2019. Iterative Dual Domain Adaptation for Neural Machine Translation. In *EMNLP 2019*.

Zeng, S.; Wu, Y.; Chang, B.; and Chang, B. 2021. SIRE: Separate Intra- and Inter-sentential Reasoning for Document-level Relation Extraction. In *ACL Findings 2021*.

Zeng, S.; Xu, R.; Chang, B.; and Li, L. 2020. Double Graph Based Reasoning for Document-level Relation Extraction. In *EMNLP 2020*.

Zhang, B.; Xiong, D.; Su, J.; and Luo, J. 2019. Future-Aware Knowledge Distillation for Neural Machine Translation. *IEEE/ACM Transactions on Audio, Speech and Language Processing*.

Zhang, L.; Su, J.; Chen, Y.; Miao, Z.; Min, Z.; Hu, Q.; and Shi, X. 2022. Towards Better Document-level Relation Extraction via Iterative Inference. In *EMNLP 2022*.

Zhang, N.; Chen, X.; Xie, X.; Deng, S.; Tan, C.; Chen, M.; Huang, F.; Si, L.; and Chen, H. 2021. Document-level Relation Extraction as Semantic Segmentation. In *IJCAI 2021*.

Zhang, Y.; Qi, P.; and Manning, C. D. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *EMNLP 2018*.

Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; and Manning, C. D. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *EMNLP 2017*.

Zhang, Z.; Yu, B.; Shu, X.; Liu, T.; Tang, H.; Yubin, W.; and Guo, L. 2020. Document-level Relation Extraction with Dual-tier Heterogeneous Graph. In *COLING 2020*.

Zhou, J.; Wei, C.; Wang, H.; Shen, W.; Xie, C.; Yuille, A.; and Kong, T. 2022. ibot: Image bert pre-training with online tokenizer. In *ICLR 2022*.

Zhou, W.; Huang, K.; Ma, T.; and Huang, J. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *AAAI 2021*.