Learning Logic Rules for Document-level Relation Extraction

Dongyu Ru[†‡], **Changzhi Sun**[‡*], **Jiangtao Feng**[‡], **Lin Qiu**[†], **Hao Zhou**[‡], **Weinan Zhang**[†*], **Yong Yu**[†], **Lei Li**[§†]
[†]Shanghai Jiao Tong University [‡]ByteDance AI Lab
[§]University of California, Santa Barbara
{maxru,lqiu,wnzhang,yyu}@apex.sjtu.edu.cn
{sunchangzhi,fengjiangtao,zhouhao.nlp}@bytedance.com
lilei@cs.ucsb.edu

Abstract

Document-level relation extraction aims to identify relations between entities in a whole document. Prior efforts to capture long-range dependencies have relied heavily on implicitly powerful representations learned through (graph) neural networks, which makes the model less transparent. To tackle this challenge, in this paper, we propose LogiRE, a novel probabilistic model for document-level relation extraction by learning logic rules. LogiRE treats logic rules as latent variables and consists of two modules: a rule generator and a relation extractor. The rule generator is to generate logic rules potentially contributing to final predictions, and the relation extractor outputs final predictions based on the generated logic rules. Those two modules can be efficiently optimized with the expectation-maximization (EM) algorithm. By introducing logic rules into neural networks, LogiRE can explicitly capture long-range dependencies as well as enjoy better interpretation. Empirical results show that LogiRE significantly outperforms several strong baselines in terms of relation performance (~1.8 F1 score) and logical consistency (over 3.3 logic score). Our code is available at https://github.com/rudongyu/LogiRE.

1 Introduction

Extracting relations from a document has attracted significant research attention in information extraction (IE). Recently, instead of focusing on sentence-level (Socher et al., 2012; dos Santos et al., 2015; Han et al., 2018; Zhang et al., 2018; Wang et al., 2021a,b), researchers have turned to modeling directly at the document level (Wang et al., 2019; Ye et al., 2020; Zhou et al., 2021), which provides longer context and requires more complex reasoning. Early efforts focus mainly on learning a powerful relation (i.e., entity pair) representation, which implicitly captures long-range dependencies. According to the input structure, we can divide the existing document-level relation extraction work into two categories: the *sequence-based model* and the *graph-based model*.

*[1] **Britain**'s Prince **Harry** is engaged to his US partner Meghan Markle. ... **[2] Harry** spent 10 years in the army and has this year, with his elder brother **William**, ... **[3]** The last major royal wedding took place In 2011, when **Kate Middleton** and Prince **William** were married.*

Entities:** **UK**, **Harry**, **William**, **Kate

Relations:** **royalty_of(Harry, UK), sibling_of(William, Harry), spouse_of(Kate, William), royalty_of(Kate, UK) ...

Rule: $royalty\_of(h,t) \leftarrow spouse\_of(h,e_1) \wedge sibling\_of(e_1,e_2) \wedge royalty\_of(e_2,t)$

Figure 1: An example of relation identification by utilizing rules. The three labeled sentences describe the relations *royalty_of*(Harry,UK), *sibling_of*(William,Harry), and *spouse_of(Kate,William)*, respectively. The identification of the relation *royalty_of*(Kate,UK) requires the synthesis of information in three sentences. It can be easily derived from the demonstrated rule and the other three relations.

The sequence-based model first leverages different sequence encoder (e.g., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019)) to obtain token representations, and then computes relation representations by various pooling operations, e.g., average pooling (Yao et al., 2019; Xu et al., 2021), attentive pooling (Zhou et al., 2021). To further capture long-range dependencies, graph-based models are proposed. By constructing a graph, words or entities that are far away can become neighbor nodes. On top of the sequence encoder, the graph encoder (e.g., GNN) can aggregate information from all neighbors, thus capturing longer dependencies. Various forms of graphs are proposed, including dependency tree (Peng et al., 2017; Zhang

[*]corresponding authors.
[†]Work is done while at ByteDance.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1239–1250
November 7–11, 2021.

et al., 2018), co-reference graph (Sahu et al., 2019), mention-entity graph (Christopoulou et al., 2019; Zeng et al., 2020), entity-relation bipartite graph (Sun et al., 2019) and so on. Despite their great success, there is still no comprehensive understanding of the internal representations, which are often criticized as mysterious "black boxes".

Learning logic rules can discover and represent knowledge in explicit symbolic structures that can be understood and examined by humans. At the same time, logic rules provide another way to explicitly capture interactions between entities and output relations in a document. For example in Fig. 1, the identification of *royalty_of(Kate,UK)* requires information in all three sentences. The demonstrated logic rule can be applied to directly obtain this relation from the three relations locally extracted in each sentence. Reasoning over rules bypasses the difficulty of capturing long-range dependencies and interprets the result with intrinsic correlations. If the model could automatically learn rules and use them to make predictions, then we would get better relation extraction performance and enjoy more interpretation.

In this paper, we propose LogiRE, a novel probabilistic model modeling intrinsic interactions among relations by logic rules. Inspired by RNN-Logic (Qu et al., 2021), we treat logic rules as latent variables. Specifically, LogiRE consists of a rule generator and a relation extractor, which are simultaneously trained to enhance each other. The rule generator provides logic rules that are used by the relation extractor for prediction, and the relation extractor provides some supervision signals to guide the optimization of the rule generator, which significantly reduces the search space. In addition, the proposed relation extractor is model agnostic, so it can be used as a plug-and-play technique for any existing relation extractors. Those two modules can be efficiently optimized with the EM algorithm. By introducing logic rules into neural networks, LogiRE can explicitly capture long-range dependencies between entities and output relations in a document and enjoy better interpretation. Our main contributions are listed below:

- We propose a novel probabilistic model for relation extraction by learning logic rules. The model can explicitly capture dependencies between entities and output relations, while enjoy better interpretation.
- We propose an efficient iterative-based method to optimize LogiRE based on the EM algorithm.
- Empirical results show that LogiRE significantly outperforms several strong baselines in terms of relation performance (~1.8 F1 score) and logical consistency (over 3.3 logic score).

2 Related Work

For document-level relation extraction, prior efforts on capturing long-range dependencies mainly focused on two directions: pursuing stronger sequence representation (Nguyen and Verspoor, 2018; Verga et al., 2018; Zheng et al., 2018) or including prior for interactions among entities as graphs (Christopoulou et al., 2019). For more powerful representations, they introduced pre-trained language models (Wang et al., 2019; Ye et al., 2020), leveraged attentions for context pooling (Zhou et al., 2021), or integrated the scattered information according to a hierarchical level (Tang et al., 2020). Aiming to model the intrinsic interactions among entities and relations, they utilized implicit reasoning structures by carefully designing graphs connecting: mentions to entities, mentions in the same sentence (Christopoulou et al., 2019; Sun et al., 2019), mentions of the same entities (Wang et al., 2020; Zeng et al., 2020), etc. Nan et al. (2020); Xu et al. (2021) directly integrated similar structural dependencies to attention mechanisms in the encoder. These approaches contributed to obtaining powerful representations for distinguishing various relations but lacked interpretability on the implicit reasoning. Another approach that can capture dependencies between relations is the global normalized model (Andor et al., 2016; Sun et al., 2018). In this work, we focus on how to learn and use logic rules to capture long-range dependencies between relations.

Another category of related work is logical reasoning. Many studies were conducted on learning or applying logic rules for reasoning. Most of them (Qu and Tang, 2019; Zhang et al., 2020) concentrated on reasoning over knowledge graphs, aiming to deduct new knowledge from existing triples. Neural symbolic systems (Hu et al., 2016; Wang and Poon, 2018) combined logic rules and neural networks to benefit from regularization on deep learning approaches. These efforts demonstrated the effectiveness of integrating neural networks with logical reasoning. Despite doc-RE providing a suitable scenario for logical reasoning (with relations serving as predicates and entities as variables),

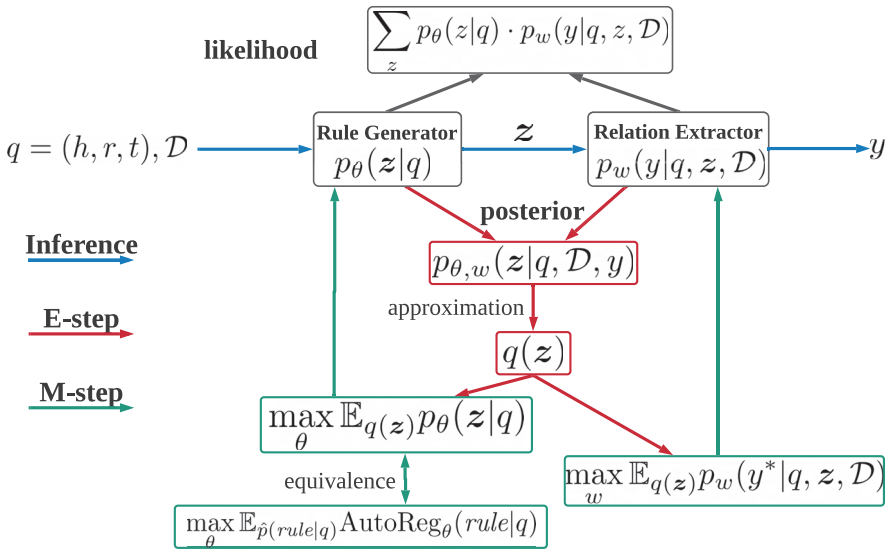

Figure 2: The overview of LogiRE. LogiRE consists of two modules: a rule generator p_θ and a relation extractor p_w. For a given document $\mathcal{D}$ and a query triple q, we treat the required logic rules as latent variables $\boldsymbol{z}$, aiming to identify the corresponding truth value y. During inference, we sample from the rule generator for the latent rule set and use the relation extractor to predict y given the rules. The overall objective (maximizing the likelihood) is optimized by the EM algorithm. In the E-step, we estimate the approximate posterior $q(\boldsymbol{z})$; In the M-step, we maximize a lower bound of the likelihood w.r.t. θ, w.

no existing work attempted to learn and utilize rules in this field. Using hand-crafted rules, Wang and Pan (2020); Wu et al. (2020) achieved great success on sent-level information extraction tasks. However, the rules were predefined and limited to low-level operations, restricting their applications.

3 Method

In this section, we describe the proposed method LogiRE that learns logic rules for document-level relation extraction. We first define the task of document-level relation extraction and logic rules.

Document-level Relation Extraction Given a set of entities $\mathcal{E}$ with mentions scattered in a document $\mathcal{D}$, we aim to extract a set of relations $\mathcal{R}$. A relation is a triple $(h,r,t) \in \mathcal{R}$ (also denoted by $r(h,t)$), where $h \in \mathcal{E}$ is the head entity, $t \in \mathcal{E}$ is the tail entity and r is the relation type describing the semantic relation between two entities. Let $\mathcal{T}_r$ be the set of possible relation types (including reverse relation types). For simplicity, we define a query $q=(h,r,t)$ and aim to model the probabilistic distribution $p(\boldsymbol{y}|q,\mathcal{D})$, where $\boldsymbol{y} \in \{-1,1\}$ is a binary variable indicating whether (h,r,t) is valid or not, and $h,t \in \mathcal{E}, r \in \mathcal{T}_r$. In this paper, bold letters indicate variables.

Logic Rule We extract relations from the document by learning logic rules, where logic rules in this work have the conjunctive form:

$$\forall\{e_i\}_{i=0}^{l}\, r(e_0,e_l) \leftarrow r_1(e_0,e_1) \wedge \cdots \wedge r_l(e_{l-1},e_l)$$

$e_i \in \mathcal{E}$, $r_i \in \mathcal{T}_r$ and l is the rule length. This form can express a wide range of common logical relations such as symmetry and transferability. For example, transferability can be expressed as

$$\forall\{e_0,e_1,e_2\}\, r(e_0,e_2) \leftarrow r(e_0,e_1) \wedge r(e_1,e_2)$$

Inspired by RNNLogic (Qu et al., 2021), to infer high-quality logic rules in the large search space, we separate rule learning and weight learning and treat the logic rules as the latent variable. LogiRE consists of two main modules: the rule generator and the relation extractor, which are simultaneously trained to enhance each other. Given the query $q=(h,r,t)$ in the document $\mathcal{D}$, on the one hand, the rule generator adopts an auto-regressive model to generate a set of logic rules based on q, which was used to help the relation extractor make the final decision; on the other hand, the relation extractor can provide some supervision signals to update the rule generator with posterior inference, which greatly reduces the search space with high-quality rules.

Unlike existing methods to capture the interactions among relations in the document by learning powerful representations, we introduce a novel probabilistic model LogiRE (Sec. 3.1, Fig. 2), which explicitly enhances the interaction by learning logic rules. LogiRE uses neural networks to parameterize the rule generator and the relation extractor (Sec. 3.2), optimized by the EM algorithm in an iterative manner (Sec. 3.3).

3.1 Overview

We formulate the document-level relation extraction in a probabilistic way, where a set of logic rules is assigned as a latent variable $\boldsymbol{z}$. Given a query variable $\boldsymbol{q} = (\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ in the document $\mathcal{D}$, we define the target distribution $p(\boldsymbol{y}|\boldsymbol{q}, \mathcal{D})$ as below[1]:

$$p_{w,\theta}(\boldsymbol{y}|\boldsymbol{q}) = \sum_{z} p_w(\boldsymbol{y}|\boldsymbol{q}, z) p_\theta(z|\boldsymbol{q})$$

where p_θ is the distribution of the rule generator which defines a prior over the latent variable $\boldsymbol{z}$ conditioned on a query $\boldsymbol{q}$ (we assume the distribution of $\boldsymbol{z}$ is independent from the document $\mathcal{D}$), and p_w is the relation extractor which gives the probability of $\boldsymbol{y}$ conditioned on the query $\boldsymbol{q}$, latent $\boldsymbol{z}$, and the document $\mathcal{D}$. Given the gold label y^* of the query q in the document $\mathcal{D}$, the objective function is to maximize the likelihood as follows:

$$\mathcal{L}(w, \theta) = \log p_{w,\theta}(y^*|q) \quad (1)$$

Due to the existence of latent variables in the objective function $\mathcal{L}$, we use the EM algorithm for optimization (Sec. 3.3).

3.2 Parameterization

We use neural networks to parameterize the rule generator and the relation extractor.

Rule Generator The rule generator defines the distribution $p_\theta(z|\boldsymbol{q})$. For a query q, the rule generator generates a set of logic rules denoted by $\boldsymbol{z}$ for predicting the truth value $\boldsymbol{y}$ of the query q.

Formally, given a query $q = (h, r, t)$, we generate logic rules that takes the form of $r \leftarrow r_1 \wedge \cdots \wedge r_l$. Such relation sequences $[r_1, \ldots, r_l]$ can be effectively modeled by an autoregressive model. In this work, we employ a Transformer-based autoregressive model AutoReg_θ to parameterize the rule generator, which sequentially generates each relation r_i. In this process, the probabilities of generated rules are simultaneously computed. Next, we assume that the rule set $\boldsymbol{z}$ obeys a multinomial distribution with N rules independently sampled from the distribution $\text{AutoReg}_\theta(rule|q)$:

$$p_\theta(\boldsymbol{z}|q) \sim \text{Multi}(\boldsymbol{z}|N, \text{AutoReg}_\theta(rule|q)),$$

where Multi denotes multinomial distribution, N is a hyperparameter for the size of the set $\boldsymbol{z}$ and AutoReg_θ defines a distribution over logic rules conditioned on the query q. [2]

Relation Extractor The relation extractor defines $p_w(\boldsymbol{y}|\boldsymbol{q}, \boldsymbol{z})$. It utilizes a set of logic rules to get the truth value of $\boldsymbol{y}$ corresponding to the query $\boldsymbol{q}$. For each query q, a $rule \in \boldsymbol{z}$ is able to find different grounding paths on the document $\mathcal{D}$. For example, Alice$\xrightarrow{father}$Bob$\xrightarrow{spouse}$Cristin is a grounding path for the rule $mother(e_0, e_2) \leftarrow father(e_0, e_1) \wedge spouse(e_1, e_2)$. Following the product t-norm fuzzy logic (Cignoli et al., 2000), we score each *rule* as follows:

$$\phi_w(rule) = \max_{path \in \mathcal{P}(rule)} \phi_w(path)$$

$$path: \underset{(h)}{e_0} \xrightarrow{r_1} e_1 \xrightarrow{r_2} e_2 \rightarrow \cdots \xrightarrow{r_l} \underset{(t)}{e_l}$$

$$\phi_w(path) = \prod_{i=1}^{l} \phi_w(e_{i-1}, r_i, e_i)$$

where $\mathcal{P}(rule)$ is the set of grounding paths which start at h and end at t following a *rule*. $\phi_w(e_{i-1}, r_i, e_i)$ is the confidence score obtained by any existing relation models. [3]

To get the probability (fuzzy truth value) of $\boldsymbol{y}$, we synthesize the evaluation result of each rule in the latent rule set $\boldsymbol{z}$. The satisfaction of any rule body will imply the truth of $\boldsymbol{y}$. Accordingly, we take the disjunction of all rules in $\boldsymbol{z}$ as the target truth value. Following the principled sigmoid-based fuzzy logic function for disjunction (Sourek et al., 2018; Wang and Pan, 2020), we define the fuzzy truth value as:

$$p_w(\boldsymbol{y}|q, \boldsymbol{z}) = \text{Sigmoid}(\boldsymbol{y} \cdot \texttt{score}_w(q, \boldsymbol{z}))$$

$$\texttt{score}_w(q, \boldsymbol{z}) = \phi_w(q) + \sum_{rule \in \boldsymbol{z}} \phi_w(q, rule) \phi_w(rule)$$

where $\phi_w(q)$ and $\phi_w(q, rule)$ are learnable scalar weights. $\phi_w(q)$ is a bias term for balancing the score of positive and negative cases.

[1] For simplicity, we omit $\mathcal{D}$ in distributions $p_{w,\theta}$ and p_w.

[2] The generative process of a rule set $\boldsymbol{z}$ is quite intuitively similar to a translation model, and we simply generate N rules with AutoReg_θ to form $\boldsymbol{z}$.

[3] This is why our approach is plug-and-play.

$\phi_w(q, rule)$ estimates the score, namely, the quality of a specific rule. $\phi_w(rule)$ evaluates the accessibility from the head entity h to the tail entity t through the meta path defined by *rule*'s body. Applying logic rules and reasoning over the rules enable the relation extractor to explicitly modeling the long-range dependencies as the interactions among entities and relations.

3.3 Optimization

To optimize the likelihood $\mathcal{L}(w, \theta)$ (Eq. 1), we update the rule generator and the relation extractor alternately in an iterative manner, namely the EM algorithm. The classic EM algorithm estimates the posterior of the latent variable $\boldsymbol{z}$ according to current parameters in the E-step; The parameters are updated in the M-step with $\boldsymbol{z}$ obeys the estimated posterior. However, in our setting, it is difficult to compute the exact posterior $p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{q})$ due to the large space of $\boldsymbol{z}$. To tackle this challenge, we seek an approximate posterior $q(\boldsymbol{z})$ by a second-order Taylor expansion. This modified version of posterior forms a lower bound on $\log p_{w,\theta}(\boldsymbol{y}|\boldsymbol{q})$, since the difference between them is a KL divergence and hence positive:

$$\overbrace{\log p_{w,\theta}(\boldsymbol{y}|\boldsymbol{q})}^{\text{log } p_{w,\theta}(\boldsymbol{y}|\boldsymbol{q})} = \overbrace{\mathop{\mathbb{E}}_{q(\boldsymbol{z})}\left[\log \frac{p_{w,\theta}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{q})}{p_{w,\theta}(\boldsymbol{z}|\boldsymbol{q}, \boldsymbol{y})}\right]}^{\text{lower bound}} - \mathop{\mathbb{E}}_{q(\boldsymbol{z})}\left[\log \frac{p_{w,\theta}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{q})}{q(\boldsymbol{z})}\right]$$
$$= \mathrm{KL}\left(q(\boldsymbol{z})||p_{w,\theta}(\boldsymbol{z}|\boldsymbol{q}, \boldsymbol{y})\right) \geq 0$$

Once we get $q(\boldsymbol{z})$, we can maximize this lower bound of $\log p_{w,\theta}(\boldsymbol{y}|\boldsymbol{q})$.

E-step Given the current parameters θ, w, E-step aims to compute the posterior of $\boldsymbol{z}$ according to the current parameters θ, w. However, the exact posterior $p_{w,\theta}(\boldsymbol{z}|\boldsymbol{q}, \boldsymbol{y})$ is nontrivial due to its intractable partition function (space of $\boldsymbol{z}$ is large). In this work, we aim to seek an approximate posterior $q(\boldsymbol{z})$.

By approximating the likelihood with the second-order Taylor expansion, we can obtain a conjugate form of the posterior as a multinomial distribution. The detailed derivation is listed in Appendix. A. Formally, we first define $H(rule)$ as the score function estimating the quality of each rule:

$$H(rule) = \log \mathrm{AutoReg}_\theta(rule|\boldsymbol{q}) + \frac{y^*}{2}\left(\frac{1}{N}\phi_w(q) + \phi_w(q, rule)\phi_w(rule)\right)$$

Intuitively, $H(rule)$ evaluates rule quality in two factors. One is based on the rule generator p_θ, which servers as the prior probability for each *rule*. The other is based on the relation extractor, and it takes into account the contribution of the current *rule* to the final correct answer y^*. Next, we use $\hat{p}(rule|q)$ to denote the posterior distribution of the *rule* given the query q:

$$\hat{p}(rule|q) \propto \exp\left(H(rule)\right)$$

Thus the approximate posterior also obeys a multinomial distribution.

$$q(\boldsymbol{z}) \sim \mathrm{Multi}\left(N, \hat{p}(rule|q)\right)$$

M-step After obtaining the $q(\boldsymbol{z})$, M-step is to maximize the lower bound $\log p_{w,\theta}(\boldsymbol{y}|\boldsymbol{q})$ with respect to both w and θ. Formally, given each data instance $(y^*, q, \mathcal{D})$ and the $q(\boldsymbol{z})$, the objective is to maximize

$$\mathcal{L}_{\text{lower}} = \overbrace{\mathop{\mathbb{E}}_{q(\boldsymbol{z})}\left[\log p_\theta(\boldsymbol{z}|q)\right]}^{\mathcal{L}_\mathrm{G}} + \overbrace{\mathop{\mathbb{E}}_{q(\boldsymbol{z})}\left[\log p_{w,\theta}(y^*|\boldsymbol{z}, q)\right]}^{\mathcal{L}_\mathrm{R}}$$

where $\mathcal{L}_\mathrm{G}, \mathcal{L}_\mathrm{R}$ are the objective of the rule generator and the relation extractor, repectively.

For the objective $\mathcal{L}_G$, it can be further converted equally as

$$\mathcal{L}_\mathrm{G} = \mathbb{E}_{\hat{p}(rule|q)}\left[\mathrm{AutoReg}_\theta(rule|q)\right]$$

To compute the expectation term of $\mathcal{L}_\mathrm{G}$ we sample from the current prior $p_\theta(\boldsymbol{z}|q)$ for a sample $\hat{z}$, and evaluate the score of each rule as $H(rule)$, normalized score over $H(rule)$ are regarded as the approximated $\hat{p}(rule|q)$. Then we use sampled rules to update the $\mathrm{AutoReg}_\theta(rule|q)$. Intuitively,

Algorithm 1 EM Optimization for $\mathcal{L}(w, \theta)$

1: **while** not converge **do**
2: For each instance, use the rule generator p_θ to generate a set of logic rules $\hat{\boldsymbol{z}}(|\hat{\boldsymbol{z}}| = N)$.
3: Calculate the rule score $H(rule)$ of each rule for approximating the posterior of *rule*: $\hat{p}(rule|\boldsymbol{q})$. ▷ E-step
4: For each instance, update the rule generator $\mathrm{AutoReg}_\theta$ based on the sampled rules from $\hat{p}(rule|\boldsymbol{q})$.
5: For each instance, update the relation extractor p_w based on generated logic rules $\hat{\boldsymbol{z}}$ from the updated rule generator. ▷ M-step
6: **end while**

Dataset		#Doc.	#Rel.	#Ent.	#Facts
DWIE	train	602	65	16494	14410
	dev	98		2785	2624
	test	99		2623	2459
DocRED	train	3053	96	59493	38180
	dev	1000		19578	12323
	test	1000		19539	-

Table 1: Statistics of Document-level RE Datasets

we update the rule generator $p_\theta(z|q)$ to make it consistent with the high-quality rules identified by the approximated posterior.

For the objective $\mathcal{L}_R$, we update the relation extractor according to the logic rules sampled from the updated rule generator. The logic rules explicitly capturing more interactions between relations can be fused as input to the relation extractor, which yields better empirical results and enjoys better interpretation. Finally, we summarize the optimization procedure in Algorithm 1.

4 Experiments

We conduct experiments on multi-relational document-level relation extraction datasets: DocRED (Yao et al., 2019) and DWIE (Zaporojets et al., 2020). The statistics of the two datasets are listed in Table 1. Pre-processing details of DWIE are described in Appendix B.

Evaluation Besides the commonly used **F1** metric for relation extraction, we also include other two metrics for comprehensive evaluation of the models: **ign F1**, **logic**. **ign F1** was proposed in (Yao et al., 2019) for evaluation with triples appearing in the training set excluded. It avoids information leakage from the training set. We propose **logic** for evaluation of logical consistency among the prediction results. Specifically, we use the 41 pre-defined rules on the DWIE dataset to evaluate whether the predictions satisfy these gold rules. The rules have a similar form to logic rules defined in Sec. 3. We name the precision of these rules on predictions as **logic** score. Note that these rules are independent of the rule learning and utilization in Sec. 3 but only used for **logic** evaluation.

Experimental Settings The rule generator in our experimental settings is implemented as a transformer with a two-layer encoder and a two-layer decoder, hidden size set to 256. We empirically find the tiny structure is enough for modeling the required rule set. We set the size of the latent rule set N to 50. We limit the maximum length of logic rules to 3 in our setting.

4.1 Baselines

We compare our LogiRE with the following baselines on document-level RE. The baselines are also used as corresponding backbone models in our framework. Yao et al. (2019) proposed to apply four state-of-the-art sentence-level RE models to document-level relation extraction: **CNN**, **LSTM**, **BiLSTM**, and **Context-Aware**. (Zeng et al., 2020) proposed **GAIN** to leverage both mention-level graph and aggregated entity-level graph to simulate the inference process in document-level RE, using graph neural networks. Zhou et al. (2021) proposed **ATLOP**, using adaptive thresholding to learn a better adjustable threshold and enhancing the representation of entity pairs with localized context pooling. The implementation details of the baselines are shown in Appendix B.

4.2 Main Results

Our LogiRE outperforms the baselines on all of the three metrics. (We mainly analyze the results on DWIE with all three metrics can be evaluated. The results on DocRED are demonstrated in Table 3 and discussed in Sec. 4.3.)

Our LogiRE consistently outperforms various backbone models. It outperforms various baselines on the DWIE dataset as shown in Table 2. We achieve 2.02 test ign F1 and 1.84 test F1 improvements on the current SOTA, ATLOP. The compatibility between LogiRE and various backbone models shows the generalization ability of our LogiRE. The consistent improvements on both sequence-based and graph-based models empirically verified the benefits of explicitly injecting logic rules to document-level relation extraction.

The improvements on graph-based models indicate the effectiveness of modeling interactions among multiple relations and entities. Despite graph-based models provide graphs (Christopoulou et al., 2019; Wang et al., 2020) consisting of connections among mentions, entities, and sentences, they seek more powerful representations which implicitly model the intrinsic connections. Our LogiRE instead builds explicit interactions among the entities and relations through the meta path determined by the rules. The improvements on the current SOTA for graph-based model empirically proved the superiority of such explicit modeling.

Model	Dev			Test		
	ign F1	F1	logic	ign F1	F1	logic
CNN	37.65	47.73	51.70	34.65	46.14	54.69
CNN + LogiRE	40.31(+2.65)	50.04(+2.71)	72.84(+21.14)	39.21(+4.65)	50.44(+4.30)	73.47(+18.78)
LSTM	40.86	51.77	65.64	40.81	52.60	61.64
LSTM + LogiRE	42.79(+1.93)	53.60(+1.83)	69.74(+4.10)	43.82(+3.01)	55.03(+2.43)	71.27(+9.63)
BiLSTM	40.46	51.92	64.87	42.03	54.47	64.41
BiLSTM + LogiRE	42.59(+2.13)	53.83(+1.91)	73.37(+8.50)	43.65(+1.62)	55.14(+0.67)	77.11(+12.70)
Context-Aware	42.06	53.05	69.27	45.37	56.58	70.01
Context-Aware + LogiRE	43.88(+1.82)	54.49(+1.44)	73.98(+4.71)	48.10(+2.73)	59.22(+2.64)	75.94(+5.93)
GAIN	58.63	62.55	78.30	62.37	67.57	86.19
GAIN + LogiRE	60.12 (+1.49)	63.91(+1.36)	**87.86** (+9.56)	**64.43** (+2.06)	69.40(+1.83)	**91.22**(+5.02)
ATLOP	59.03	64.82	81.98	62.09	69.94	82.76
ATLOP + LogiRE	**60.24**(+1.21)	**66.76**(+1.94)	86.98(+5.00)	64.11(+2.02)	**71.78**(+1.84)	86.07(+3.31)

Table 2: Main results on DWIE. (The underlined statistics pass a t-test for significance with p value < 0.01.)

Model	Test	
	ign F1	F1
GAIN	57.93	60.07
GAIN + LogiRE	58.62(+0.69)	60.61(+0.54)
ATLOP	59.14	61.13
ATLOP + LogiRE	59.48(+0.34)	61.45(+0.32)

Table 3: Comparison on DocRED. The improvements are less significant with reasons analyzed in Sec. 4.3.

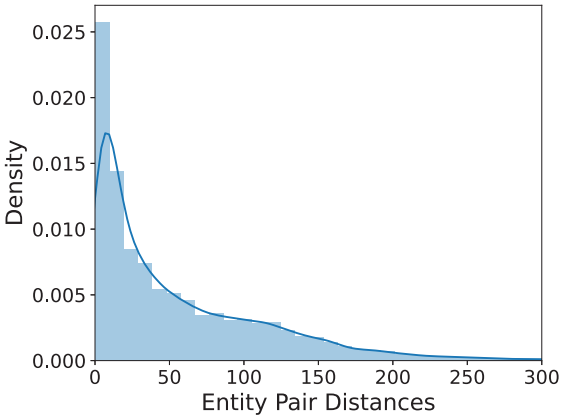

Figure 3: Distribution of Distance Between Entity Pairs in DocRED

Our model achieves better logical consistency compared with the baselines. The results show that LogiRE achieves up to 18.78 enhancement on the **logic** metric. Even on the graph-based model, GAIN, we obtain a significant improvement of 5.03 on logical consistency. The improved **logic** score shows that the predictions of LogiRE are more consistent with the regular logic patterns in the data. These numbers are evidence of the strength of our iterative-based optimization approach by introducing logic rules as latent variables.

4.3 Analysis & Discussion

We analyze the results on DocRED data and discuss the superiority of our LogiRE on capturing long-range dependencies and interpretability. The capability of capturing long-range dependencies is studied by inspecting the inference performance on entity pairs of various distances. The interpretability is verified by checking the logic rules learned by our rule generator and the case study on predictions.

Analysis on DocRED Results In comparison with the significant improvements on DWIE, the enhancement of LogiRE on DocRED is less significant. Our analysis shows that the reasons are relatively shorter dependencies in DocRED and the logical inconsistency caused by incomplete annotations.

a) Shorter Dependencies in DocRED Shorter dependencies in DocRED lower the demand for capturing long-range correlations among entities and relations. We show the distribution of distance between entity pairs in Fig. 3. 79.26% of entity pairs in DocRED have distances less than 100 tokens. The examples in DocRED are less difficult on capturing long-range dependencies. More analysis and comparison can be found in Zaporojets et al. (2020). The representation-based approaches can already perform well in such cases. The benefits of modeling long-range dependencies through logical reasoning will be smaller.

b) Logical Inconsistency in DocRED The justification of predictions after reasoning may be not accurate because of missing annotations. We calculated the error rate of a few easy-to-verify logic rules as shown in Table. 4. The 7 rules, selected

implication rule	error rate
$\text{father}(h,z) \wedge \text{spouse}(z,t) \rightarrow \text{mother}(h,t)$	24.07%
$\text{replaces}^{-1}(h,t) \rightarrow \text{replaced\_by}(h,t)$	22.22%
$\text{capital}^{-1}(h,t) \rightarrow \text{capital\_of}(h,t)$	28.24%
$\text{father}^{-1}(h,t) \rightarrow \text{child}(h,t)$	10.26%
$\text{followed}^{-1}(h,t) \rightarrow \text{follows}(h,t)$	22.40%
$\text{capital}^{-1}(h,t) \rightarrow \text{capital\_of}(h,t)$	28.24%
$\text{P150}^{-1}(h,t) \rightarrow \text{P131}(h,t)$	19.71%

Table 4: The logical inconsistency in the DocRED (for conciseness, P150 represents the relation 'contains administrative territorial entity' and P131 represents the relation 'located in the administrative territorial entity'). The shown easy-to-verify gold rules have high error rates in DocRED while a considerable part of relations (12.96%) are involved in as atoms in shown rules. Those missing annotations make the learning of logic rules difficult. Inconsistent patterns or statistics between training and test may lead to unfair evaluation of relation extraction performance.

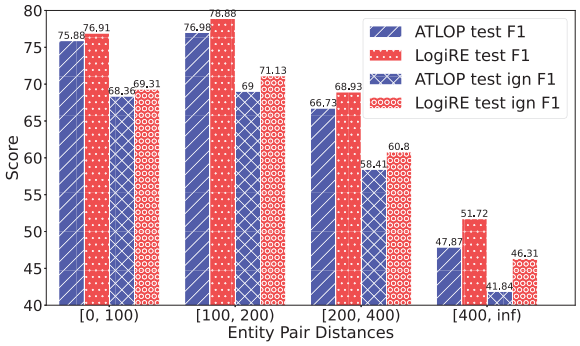

Figure 4: Performance gaps between ATLOP and LogiRE-ATLOP for entity pairs with different distances.

by case study, have a considerable part (12.96%) of labeled relations may participate in as atoms. However, the statistics in the table demonstrated that all the 7 rules have error rates higher than 10%. The numbers indicated that a notable partition of true relations are missing. The results obtained by reasoning over logic rules may be wrongly justified since the data is not exhaustively annotated.

According to the analysis above, our LogiRE has greater potential than that demonstrated as the overall performance on DocRED.

Logic rules are shortcuts for comprehension. The performance enhancement of our LogiRE becomes more prominent when the distance between entity pairs gets longer. We plot the performance of ATLOP and ATLOP-based LogiRE on the DWIE dataset with four groups of entity pair distances in Fig. 4. The distance is calculated as the number of tokens in between the nearest mentions of an entity pair. Results indicate that our LogiRE performs better on capturing long dependencies.

$\text{played\_by}(h,z) \wedge \text{plays\_in}(z,t) \rightarrow \text{character\_in}\ (h,t)$
$(\text{parent\_of}(h,z) \vee \text{child\_of}(h,z) \vee \text{spouse\_of}(h,z))$ $\wedge\ \text{royalty\_of}(z,t) \rightarrow \text{royalty\_of}(h,t)$
$\text{event\_in2}^{-1}(h,z) \wedge \text{event\_in0}(z,t) \rightarrow \text{in0}(h,t)$
$\text{minister\_of}(h,z) \wedge \text{in0}(z,t) \rightarrow \text{citizen\_of}(h,t)$
$\text{member\_of}^{-1}(h,z) \wedge \text{agent\_of}(z,t) \rightarrow \text{based\_in0}(h,t)$

Table 5: Example rules extracted from LogiRE trained on the DWIE dataset.

Relation extraction for entity pairs with longer distances in between generally performs worse. As shown in the figure, the performance starts to drop as the distance surpasses 100 tokens, indicating the difficulty of modeling long-range dependencies. The redundant information in a long context impedes accurate semantic mapping through powerful representations. This issue increases the complexity of modeling and limits the potential of representation-based approaches.

Our framework with latent logic rules injected can effectively alleviate this problem. The performance drop of our LogiRE is smaller when the distance between entities gets larger. For entity pairs of distances larger than 400, our LogiRE achieves up to 4.47 enhancement on test ign F1. By reasoning over local logic units (atoms in rules), we ignore the noisy background information in the text but directly integrate high-level connections among concepts to get the answer.

The reasoning process of our LogiRE is in line with the comprehension way of we human beings when reading long text. We construct basic concepts and connections between (local logic atoms) for each local part of the text. When the collected information is enough to fit some prior knowledge (logic rules), we deduct new cognition from the existing knowledge. Our LogiRE provides shortcuts for modeling long text semantics by adding logic reasoning to naive semantics mapping.

Interpretability by Generating Rules Our LogiRE enjoys better interpretability with the generated latent rule set. After the EM optimization, we can sample from the rule generator for high-quality rules that may contribute to the final predictions. Besides the gold rules previously shown for evaluating **logic**, LogiRE mines more logic rules from the data, as shown in Table. 5. These logic rules explicitly reveal the interactions among entities and

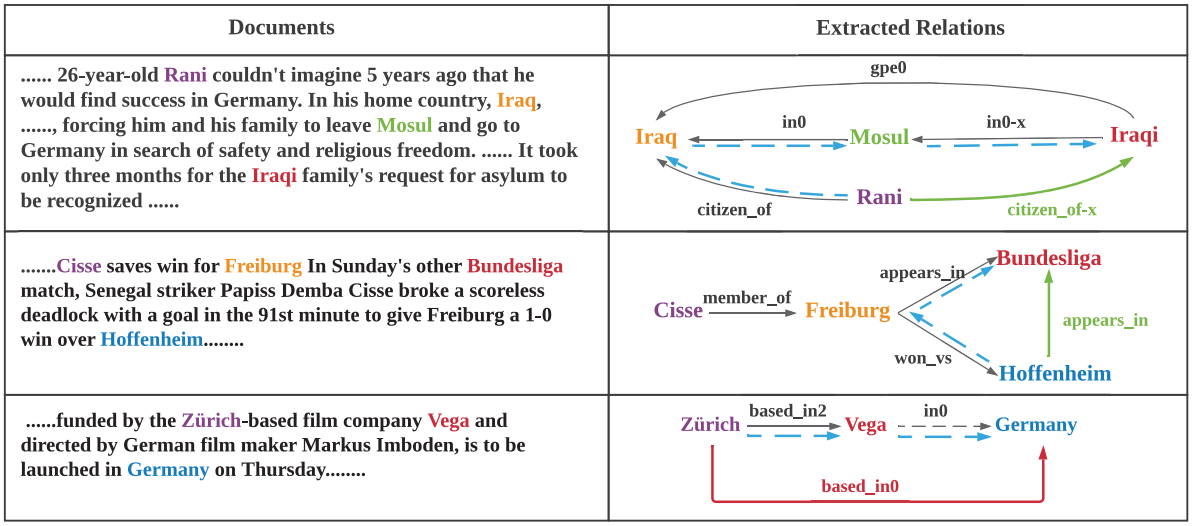

Figure 5: Inference cases of our LogiRE on DWIE by using ATLOP as the backbone model. The grey arrows are relations extracted by the backbone model, solid lines representing true relations while dashed lines representing false relations. The green arrows are new relations correctly extracted by logical reasoning. The blue arrows indicate the potential reasoning paths. We also demonstrate a negative case. In the third example, the red arrow represents a wrong relation extracted by reasoning over wrongly estimated atoms.

relations in the same document as regular patterns. LogiRE is more transparent, exhibiting the latent rules by the rule generator.

Case Study Fig. 5 shows a few inference cases of our LogiRE, including two positive examples and a negative one. As shown in the first two examples, LogiRE can complete the missing relations in the backbone model's outputs by utilizing logical rules. The soft logical reasoning can remedy the defects of representation-based approaches under specific circumstances. However, the extra reasoning may also exacerbate errors by reasoning over wrongly estimated logic units. The third example shows such a case. The wrongly estimated atom *in0*(Vega, Germany) leads to one more wrong relation extracted by reasoning. Fortunately, such errors in our LogiRE will be more controllable because of the transparency in the logical reasoning part.

5 Conclusion

In this paper, we proposed a probabilistic model LogiRE, which utilizes rules and conducts reasoning over the rules for document-level relation extraction. The logic rules are treated as latent variables. We utilize the EM algorithm to efficiently maximize the overall likelihood. By injecting rules to the relation extraction framework, our LogiRE explicitly models the long-range dependencies in docRE as interactions among relations and entities, thus enjoying better interpretability. Empirical results and analysis show that LogiRE outperforms strong baselines on overall performance, logical consistency, and capability for capturing long-range dependencies.

Acknowledgements

The SJTU team is supported by Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and National Natural Science Foundation of China (61772333). We thank Xinbo Zhang, Jingjing Xu, Yuxuan Song, Wenxian Shi and other anonymous reviewers for their insightful and detailed comments.

References

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4925–4936, Hong Kong, China. Association for Computational Linguistics.

Roberto Cignoli, Francesc Esteva, Lluís Godo, and Antoni Torrens. 2000. Basic fuzzy logic is the logic of continuous t-norms and their residua. *Soft computing*, 4(2):106–112.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Beijing, China. Association for Computational Linguistics.

Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. 2018. Hierarchical relation extraction with coarse-to-fine grained attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2245, Brussels, Belgium. Association for Computational Linguistics.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557, Online. Association for Computational Linguistics.

Dat Quoc Nguyen and Karin Verspoor. 2018. Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings. In *Proceedings of the BioNLP 2018 workshop*, pages 129–136, Melbourne, Australia. Association for Computational Linguistics.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. 2021. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *International Conference on Learning Representations*.

Meng Qu and Jian Tang. 2019. Probabilistic logic neural networks for reasoning. In *Advances in neural information processing systems*.

Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Inter-sentence relation extraction with document-level graph convolutional neural network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4309–4316, Florence, Italy. Association for Computational Linguistics.

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, Jeju Island, Korea. Association for Computational Linguistics.

Gustav Sourek, Vojtech Aschenbrenner, Filip Zelezny, Steven Schockaert, and Ondrej Kuzelka. 2018. Lifted relational neural networks: Efficient learning of latent relational structures. *Journal of Artificial Intelligence Research*, 62:69–100.

Changzhi Sun, Yeyun Gong, Yuanbin Wu, Ming Gong, Daxin Jiang, Man Lan, Shiliang Sun, and Nan Duan. 2019. Joint type inference on entities and relations via graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1361–1370.

Changzhi Sun, Yuanbin Wu, Man Lan, Shiliang Sun, Wenting Wang, Kuang-Chih Lee, and Kewen Wu. 2018. Extracting entities and relations with joint minimum risk training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2265.

Hengzhu Tang, Yanan Cao, Zhenyu Zhang, Jiangxia Cao, Fang Fang, Shi Wang, and Pengfei Yin. 2020. Hin: Hierarchical inference network for document-level relation extraction. *Advances in Knowledge Discovery and Data Mining*, 12084:197.

Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 872–884, New Orleans, Louisiana. Association for Computational Linguistics.

Difeng Wang, Wei Hu, Ermei Cao, and Weijian Sun. 2020. Global-to-local neural networks for

document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3711–3721, Online. Association for Computational Linguistics.

Hai Wang and Hoifung Poon. 2018. Deep probabilistic logic: A unifying framework for indirect supervision. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1891–1902, Brussels, Belgium. Association for Computational Linguistics.

Hong Wang, Christfried Focke, Rob Sylvester, Nilesh Mishra, and William Wang. 2019. Fine-tune bert for docred with two-step process. *arXiv preprint arXiv:1909.11898*.

Wenya Wang and Sinno Jialin Pan. 2020. Integrating deep learning with logic fusion for information extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9225–9232.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021a. ENPAR:enhancing entity and entity pair representations for joint entity relation extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2877–2887, Online. Association for Computational Linguistics.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021b. UniRE: A unified label space for entity relation extraction. 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 220–231, Online. Association for Computational Linguistics.

Meixi Wu, Wenya Wang, and Sinno Jialin Pan. 2020. Deep Weighted MaxSAT for Aspect-based Opinion Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5618–5628, Online. Association for Computational Linguistics.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online. Association for Computational Linguistics.

Klim Zaporojets, Johannes Deleu, Chris Develder, and Thomas Demeester. 2020. Dwie: an entity-centric dataset for multi-task document-level information extraction. *arXiv preprint arXiv:2009.12626*.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640, Online. Association for Computational Linguistics.

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

Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. 2020. Efficient probabilistic logic reasoning with graph neural networks. In *International Conference on Learning Representations*.

Wei Zheng, Hongfei Lin, Zhiheng Li, Xiaoxia Liu, Zhengguang Li, Bo Xu, Yijia Zhang, Zhihao Yang, and Jian Wang. 2018. An effective neural model extracting document level chemical-induced disease relations from biomedical literature. *Journal of biomedical informatics*, 83:1–9.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Approximation of the True Posterior

The exact posterior of the latent rule set $\boldsymbol{z}$ is difficult to be directly calculated because of the large space. In this section, we provide the detailed derivation for the approximate posterior.

$$
\begin{aligned}
&\log p(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{q},\mathcal{D}) \\
=&\log p_w(\boldsymbol{y}|\boldsymbol{q},\boldsymbol{z},\mathcal{D}) + \log p_\theta(\mathbf{z}|\mathbf{q}) + C \\
=&\log \frac{1}{1+e^{-\boldsymbol{y}\cdot \texttt{score}(\mathbf{q},\mathbf{z})}} + \sum_{z\in\boldsymbol{z}} \log p_\theta(z|\boldsymbol{q}) + C \\
\approx&\frac{1}{2}\boldsymbol{y}\cdot \texttt{score}_w(\boldsymbol{q},\boldsymbol{z}) + \sum_{z\in\boldsymbol{z}} \log p_\theta(z|\boldsymbol{q}) + C \\
=&\sum_{rule\in\boldsymbol{z}} (\frac{1}{2}\boldsymbol{y}\cdot(\frac{1}{N}(\phi_w(\boldsymbol{q}) + \phi_w(\boldsymbol{q},rule)\phi_w(rule)) \\
&+ \log \text{AutoReg}_\theta(rule|\boldsymbol{q})) + C
\end{aligned}
$$

The approximation is obtained by the following second-order Taylor expansion:

$$-\log(1+e^{-x}) = -\log 2 + \frac{x}{2} + O(x^2)$$

By such approximation, we can decompose the posterior to each rule in the latent rule set. We first define the score for each rule:

$$H(rule) = \log \text{AutoReg}_\theta(rule|\boldsymbol{q}) + \frac{y^*}{2}\left(\frac{1}{N}\phi_w(q) + \phi_w(q, rule)\phi_w(rule)\right)$$

Then, it's easy to obtain that the approximated posterior $q(z)$ and the prior p_θ are conjugate distributions.

$$q(z) \sim \text{Multi}(N, \frac{1}{Z}\exp(H(rule)))$$

where Z is the normalization factor.

B Implementation Details

DWIE Dataset Preprocessing The original DWIE dataset (Zaporojets et al., 2020) is designed for four sub-tasks in the information extraction, including named entity recognition, coreference resolution, relation extraction, and entity linking. In this paper, we focus on the document-level RE task. We only use the dataset for document-level relation extraction. The original dataset published 802 documents with 23130 entities in total, 702 for train and 100 for test. In our setting, we remove the entities without mentions in the context. After the cleaning, we have 700 documents for train and 99 documents for test. The training set is then randomly split into two parts: 602 documents for train and 98 for development. The statistics of the preprocessed dataset are shown in Table 1 of the main body.

Baselines We use their published open-source code to implement the baselines (Yao et al., 2019; Zeng et al., 2020; Zhou et al., 2021), as well as the backbone models in our framework. The pre-trained language models used in GAIN and ATLOP follows the original paper (Zeng et al., 2020; Zhou et al., 2021), using the pre-trained bert-base-uncased and bert-base-cased models respectively. The hyperparameters reserve the same as in their papers.