From What to Why: Improving Relation Extraction with Rationale Graph

Zhenyu Zhang, Bowen Yu, Xiaobo Shu, Mengge Xue, Tingwen Liu* and **Li Guo**
Institute of Information Engineering, Chinese Academy of Sciences. Beijing, China
School of Cyber Security, University of Chinese Academy of Sciences. Beijing, China
{zhangzhenyu1996, yubowen, shuxiaobo}@iie.ac.cn
{xuemengge, liutingwen, guoli}@iie.ac.cn

Abstract

Which type of information affects the existing neural relation extraction (RE) models to make correct decisions is an important question. In this paper, we observe that entity type and trigger are the most indicative information for RE in each instance. Moreover, these indicative clues are always constrained to co-occur with specific relations at the corpus level. Motivated by this, we propose a novel RAtionale Graph (RAG) to organize such co-occurrence constraints among entity types, triggers and relations in a holistic graph view. By introducing two subtasks of entity type prediction and trigger labeling, we build the connection between each instance and RAG, and then leverage relevant global co-occurrence knowledge stored in the graph to improve the performance of neural RE models. Extensive experimental results indicate that our method outperforms strong baselines significantly and achieves state-of-the-art performance on the document-level and sentence-level RE benchmarks.

1 Introduction

Relation extraction (RE), which aims to identify the semantic relation between two entities in plain text, is one of the fundamental tasks in information extraction (IE). In the deep learning era, many approaches are proposed including models based on attention mechanism (Lin et al., 2016; Zhang et al., 2017), graph neural networks (Zhang et al., 2018; Guo et al., 2019), and pre-trained language models (Joshi et al., 2020; Yu et al., 2020).

While these neural RE models have achieved the latest state-of-the-art results, little is known about which type of information affects the models to make decisions. Recently, an empirical study shows that the understanding of two main information sources, entity type, and textual context, is necessary and effective for training a RE model (Peng et al., 2020). Entity type, is always an important side information for RE (Liu et al., 2014; Vashishth et al., 2018). In the textual context, some words play an indicative role in relation expression. Yu et al. (2020) initially annotated the minimal contiguous indicative word span and named them *trigger*. For example, in Figure 1, when we notice that both the subject and object entities are `person`, as well as the trigger `children` appears in the context, our immediate reaction is that they probably hold a parent-child relation, then we make a further judgment by reading the complete text.

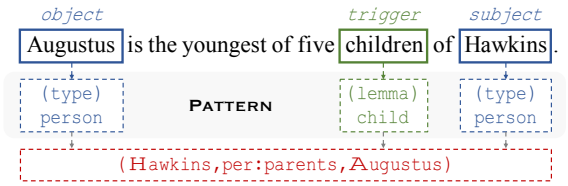

Figure 1: Illustration of the decision-making process in RE, where patterns are the most indicative information.

What is the support behind such rapid and accurate decision-making of human beings? In RE, if we look at the entire corpus from a global view, we can find a common phenomenon that one certain entity type or trigger is constrained to co-occur with specific relations. Taking entity type as an example, two entities of type `person` can only participate in person-related relations (e.g., `per:parents`, `per:siblings`). Such global co-occurrence induced by multiple seen instances serves as the crucial prior knowledge in the process of human cognition (Chater et al., 2006), and can naturally form a bipartite graph, in which the nodes on two sides are entity types and relations respectively. Similarly, the same logic can also go for triggers.

Inspired by the above observation, in this paper, we propose a RAtionale Graph (RAG) to organize the global co-occurrence statistics aggregated from

* Corresponding Author.

the corpus. Specifically, nodes in the graph are constructed based on the relations and patterns[1]. There are totally four types of directed edges that exist between different types of nodes. For example, the edge between a trigger node and a relation node depicts the co-occurrence probability of a text expressing the relation when the trigger appears in the text. This probabilistic knowledge, together with the involved nodes, is collectively referred to as *rationale*. In the end, RAG is excepted to present a holistic view of all patterns and relations, and then facilitate the relation prediction.

Now we incorporate RAG with neural networks to improve the RE performance. Given an instance with a text and two entities, we first predict the entity type and label the trigger, then establish the link between the input instance with the known patterns in RAG, and finally enhance the instance representation with the attended relation node features in the graph. Meanwhile, we introduce the gate mechanism and graph neural networks (GNNs) to perform the information propagation from the input instance to relation nodes. Hence, this workflow makes full use of all aforementioned rationale knowledge to guide the processing of new instances by linking them to each seen pattern stored in the graph, like humans recognizing new things by intuitively associating with the knowledge they have memorized. In the training phase, the model learns simultaneously (1) the relation along with (2) the entity type and trigger for each instance. This means that we care about not only the final relation label (*what*), but also the intermediate results, i.e., whether the entity type and trigger are correctly predicted (*why*). By doing so, we can retrieve the relevant global pattern knowledge from the graph with the predicted trigger and entity types, during testing.

To evaluate our approach, we first conduct experiments on the document-level RE task DialogRE (Yu et al., 2020). Experimental results show the benefits of the proposed method, leading to state-of-the-art performance. An exciting discovery is that our method is very effective in small-scale annotation scenes, using only half (with 2,584 positive instances) of the pattern-annotated instances results in a comparable performance as using all conventional annotated instances. To further validate this advantage, we manually annotate 20% (with 2,585 positive instances) patterns of the sentence-level RE benchmark TACRED (Zhang et al., 2017), and empirically demonstrate similar experimental conclusions with DialogRE.

[1]For the sake of generality, we refer to the *entity type* and *trigger* as *pattern* in the remaining of this paper.

2 Related Work

Extracting relational facts between entities from text is an essential and classical problem in natural language processing. The popular research methods have gone through the iteration from pattern-based methods (Mooney, 1999; Chang and Lui, 2001) to feature-based methods (Kambhatla, 2004; Zhou et al., 2005), and then to neural-based methods (Zeng et al., 2014; Zhang et al., 2017). Nowadays, most state-of-the-art work develops powerful neural models based on pre-trained language models or graph neural networks (Soares et al., 2019; Zhang et al., 2019; Guo et al., 2019). All the time, there are two main consensuses in the community: when extracting a relation, entity types are important side indicators, which are often used to enhance the input or output layer (Vashishth et al., 2018; Kuang et al., 2020). On the other hand, not all the words in the text are beneficial to RE. Thus there are also efforts focusing on the heuristic or implicit selection of the key clues related to relation expression (Zhang et al., 2018; Yu et al., 2019), and Yu et al. (2020) is the first work to annotate such clue words in texts and name them trigger.

However, most previous studies are only based on local features, in other words, models are trained on individual instance, limiting the ability to capture the connection between textual indicative information and relations globally. Conversely, Su et al. (2018) emphasized the importance of the global view, and embed the textual relations with global statistics to combat the wrong labeling problem of distant supervision. Wang et al. (2020) proposed an interpretable network embedding model based on a corpus-level entity graph to rationalize medical relation prediction. Unfortunately, their methods are not suitable for the supervised RE task in the general domain. The most related work, (Zhang et al., 2020), collected a global type-relation mapping as prior knowledge to guide the optimization with knowledge distillation. One major difference is that we systematically consider both entity type and textual trigger to collect all indicative knowledge in a holistic view. Another unique aspect of this work is that we perform the prediction of entity type and trigger as two subtasks, while previous studies only focus on the final relation labels.

3 Rationale Graph (RAG)

Different from existing work only using raw text for RE, we assume the global co-occurrence statistics among relations, triggers, and entity types is given, which are pre-construed based on the whole corpus, and denoted as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each vertex $v \in \mathcal{V}$ refers the *relation*, *trigger*, or *entity type pair* extracted from the corpus and each edge $e \in \mathcal{E}$ is associated with the global co-occurrence count for the connected nodes. Inspired by Zhang et al. (2020), we organize the global co-occurrence count between two kinds of nodes as bipartite rationale mapping and pack all bipartite mappings together to obtain a rationale graph (RAG). Figure 2 shows the schematic diagram for clarity.

3.1 Bipartite Rationale Mapping

Here we take *type* (short for *entity type pair*) and *relation* as an example to describe the construction process of bipartite rationale mapping. Specifically, for instance with a text x and two entities (s, o), we combine two entity types to achieve a *pattern* p. From this step, we obtain the pattern set $\mathcal{T} = \{t_i\}$ and formulate a support set $\mathcal{S}(t_i)$ for each t_i, in which the support set $\mathcal{S}(t_i)$ contains all instances with pattern t_i. Besides, we also collect a set of relations $\mathcal{R} = \{r_j\}$, and the support set $\mathcal{S}(r_j)$ denoting the set of instances holding relation r_j. The co-occurrence number of pattern t_i and relation r_j is defined as $w_{ij} = |\mathcal{S}(t_i) \cap \mathcal{S}(r_j)|$. In other word, every instance (x, s, o) with pattern t_i and relation r_j is counted as a co-occurrence of t_i and r_j.

However, it is inappropriate to take the raw co-occurrence count as mapping weight directly. The relation distribution in reality typically has a power-law tail (Zhang et al., 2017), meaning that the count spans several orders of magnitude in different relations. To meet this challenge, for each pattern, we normalize its co-occurrence count to form a valid probability distribution over relations. In the end, the bipartite mapping $\mathcal{M}_{tp2re}$ is constructed, with one node set being the types, the other being the relations, and the weighted edges $\bar{w}_{ij} = p(r_j|t_i) = w_{ij} / \sum_{j'} w_{ij'}$ representing the normalized global co-occurrence probability.

3.2 Graph Construction

Considering that trigger and type are two kinds of information sources for RE (Peng et al., 2020), we first introduce the bipartite rationale mapping from type to relation $\mathcal{M}_{tp2re}$ and the mapping from trigger to relation $\mathcal{M}_{tg2re}$ in RAG. In this way, we assume that the graph reflects the prior probability of relation when some indicative information appears in the text. Furthermore, triggers are actually relations in the form of natural language (Hu et al., 2020) and entity types are tightly bound to certain trigger words within the context (Lin et al., 2020). In other words, type and trigger are mutually related and restricted. Therefore, we introduce a set of bidirectional mapping, that is, from type to trigger $\mathcal{M}_{tp2tg}$ and from trigger to type $\mathcal{M}_{tg2tp}$. Finally, we place four kinds of edges in the graph: $\mathcal{E} \leftarrow \{\mathcal{M}_{tp2re}, \mathcal{M}_{tg2re}, \mathcal{M}_{tp2tg}, \mathcal{M}_{tg2tp}\}$[2].

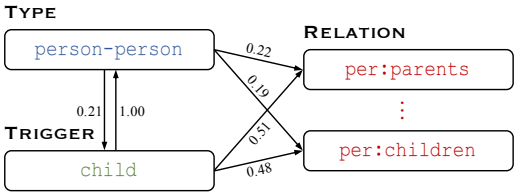

Figure 2: Schematic diagram of RAG, in which edges are weighted by normalized co-occurrence statistics.

4 Relation Extraction with RAG

In this section, we exemplify how to incorporate existing RE models with RAG. Given a text, a subject entity, and an object entity, the model aims to identify the semantic relationship between these two entities with the aid of RAG. Moreover, we also require the model to predict entity type pair and label trigger (if possible) as two auxiliary subtasks. For the example in Figure 3, we build a unified model that not only accurately predicts the relation `per:parents`, but also provides meaningful rationales on how the prediction is made: the subject and object entities are both `person`, and the key clue `children` appears in the context.

4.1 Encoding Module

We utilize BERT (Devlin et al., 2019) as the feature encoder to extract token representations due to its effectiveness in representation learning. Theoretically, the encoding module can be easily replaced by other advanced models. The encoder receives a BERT-style packed sequence and outputs a context representation matrix $\mathbf{H} \in \mathbb{R}^{n \times d}$ with an overall vector $\mathbf{h}_{cls} \in \mathbb{R}^d$ (the representation of the [CLS] token in BERT), where d is the vector dimension

[2]In view of the diversity of natural language, we use spaCy (`https://spacy.io/`) to perform lemmatization on triggers, before putting them into RAG as vertexes.

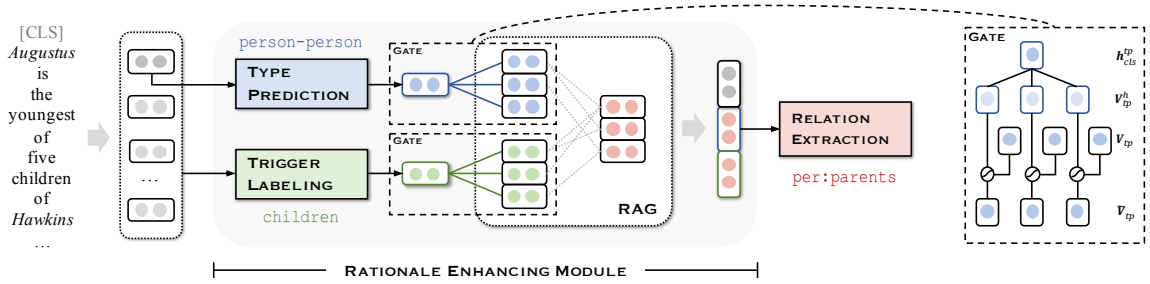

Figure 3: The overall architecture of the proposed model. Rationale enhancing module is the core component in our approach, which enhances the instance representation by retrieving pertinent rationales stored in RAG.

of the last layer of BERT. Typically, existing BERT-based RE solutions first concatenate target entities with the text or mark them in the input sequence with special tokens, and then directly take $\mathbf{h}_{cls}$ as the input of final classification module (Joshi et al., 2020; Yu et al., 2020).

4.2 Rationale Enhancing Module

The rationale enhancing module consists of two enhancing branches and one rationale integration unit. In each branch, we first predict pattern (type or trigger) for the input instance and then calculate the *pattern probability* that the instance belongs to each pattern in RAG. The integration unit aims to collect rationale enhancing features for final relation extraction based on the pattern probability and the rationale in the graph.

4.2.1 Type Enhancing Branch

In this branch, we predict the types of subject and object entities at the same time. Similar to RE, type prediction is regarded as a closed-world classification problem, and the class space is all seen entity type pairs, that is, all type nodes in RAG. Following the classification paradigm of BERT (Devlin et al., 2019), we project the overall vector $\mathbf{h}_{cls}$ into a new space for type prediction:

$$\begin{aligned} \mathbf{h}_{cls}^{tp} &= \texttt{tanh}(\texttt{MLP}_{\{d,d\}}(\mathbf{h}_{cls})), \\ \mathbf{p}_{tp} &= \texttt{SoftMax}\left(\texttt{MLP}_{\{d,n_{tp}\}}(\mathbf{h}_{cls}^{tp})\right). \end{aligned} \tag{1}$$

Here $\texttt{MLP}_{d,n_{tp}}(\cdot)$ denotes a multi-layer perceptron module with input dimension d and output dimension n_{tp}, $\mathbf{p}_{tp} \in \mathbb{R}^{n_{tp}}$ is the type probability that the given instance belongs to each type pair, where n_{tp} is the number of all known type pairs.

4.2.2 Trigger Enhancing Branch

Different from the prediction of entity type, triggers are flexible and can be any word or phrase in the text. We formulate the trigger recognition task as a labeling problem with two label sequences.

Given the representation matrix $\mathbf{H}$ output from BERT, the model predicts two probabilities of each token being the start index and end index of a trigger, respectively. To handle the instances without clear trigger (about half of them), we concatenate $\mathbf{H}$ with $\mathbf{h}_{cls}$ to form $\bar{\mathbf{H}} = [\mathbf{H}; \mathbf{h}_{cls}]$, and set the boundary index pointing to the [CLS] token. These two probability distributions over the entire sequence $\mathbf{p}_{sta}, \mathbf{p}_{end} \in \mathbb{R}^{(n+1)}$ can be obtained by

$$\begin{aligned} \mathbf{p}_{sta} &= \texttt{SoftMax}\left(\texttt{MLP}_{\{d,1\}}(\bar{\mathbf{H}})\right), \\ \mathbf{p}_{end} &= \texttt{SoftMax}\left(\texttt{MLP}_{\{d,1\}}(\bar{\mathbf{H}})\right). \end{aligned} \tag{2}$$

To align the labeling result with the triggers in RAG, we first weight each token in $\bar{\mathbf{H}}$ based on the two index probabilities and get the representation of predicted trigger $\mathbf{h}_{pre}^{tg} \in \mathbb{R}^d$, then calculate and normalize the similarity between $\mathbf{h}_{pre}^{tg}$ and all known triggers $\mathbf{V}_{tg} \in \mathbb{R}^{n_{tg} \times d}$:

$$\begin{aligned} \mathbf{h}_{pre}^{tg} &= \frac{1}{2}\left(\mathbf{p}_{sta} + \mathbf{p}_{end}\right)\bar{\mathbf{H}}, \\ \mathbf{p}_{tg} &= \texttt{SoftMax}\left(\texttt{sim}(\mathbf{h}_{pre}^{tg}, \mathbf{V}_{tg})\right), \end{aligned} \tag{3}$$

where $\mathbf{p}_{tg} \in \mathbb{R}^{n_{tg}}$ is the probability of the given instance corresponding to each known trigger, n_{tg} is the number of all triggers, and $\texttt{sim}(\cdot)$ is a similarity function as follows:

$$\begin{aligned} \texttt{sim}(\mathbf{h}_{pre}^{tg}, \mathbf{v}_{tg}^{i}) = &\texttt{MLP}_{\{4d,1\}}([\mathbf{h}_{pre}^{tg}; \mathbf{v}_{tg}^{i}; \\ &\mathbf{h}_{pre}^{tg} - \mathbf{v}_{tg}^{i}; \mathbf{h}_{pre}^{tg} \circ \mathbf{v}_{tg}^{i}]), \end{aligned} \tag{4}$$

where $\mathbf{v}_{tg}^{i} \in \mathbb{R}^d$ is the i-th trigger in $\mathbf{V}_{tg}$ and $\circ$ denotes element-wise product. In that case, even if we run into a new trigger that we have never seen before, we can also estimate the correlation between the new trigger and the known triggers via semantic similarity, and then absorb more global statistics from similar triggers. It provides the possibility for the rationale enhancing on trigger branch.

4.2.3 Rationale Integration

For each type node in RAG, we update its embedding with the instance type feature $\mathbf{h}_{cls}^{tp}$. It is intuitive that the higher the probability of an instance to a type, the more its contribution to the updating process of that type. Specifically, we first compute the update representation for each type node based on the pattern probability $\mathbf{p}_{tp}$, and then aggregate information on the text side $\mathbf{V}_{tp}^{h} \in \mathbb{R}^{n_{tp} \times d}$ and graph side $\mathbf{V}_{tp} \in \mathbb{R}^{n_{tp} \times d}$ via a gate mechanism:

$$\begin{aligned} \mathbf{V}_{tp}^{h} &= \mathbf{p}_{tp}^{\top} \mathbf{h}_{cls}^{tp}, \\ \delta_{tp} &= \texttt{Sigmoid}\left(\texttt{MLP}_{\{2d,1\}}([\mathbf{V}_{tp}; \mathbf{V}_{tp}^{h}])\right), \\ \tilde{\mathbf{V}}_{tp} &= (1 - \delta_{tp}) \circ \mathbf{V}_{tp} + \delta_{tp} \circ \mathbf{V}_{tp}^{h}. \end{aligned} \tag{5}$$

Similarly, we perform the same computation on the trigger branch to reconstruct the trigger node embeddings in RAG and result in $\bar{\mathbf{V}}_{tg} \in \mathbb{R}^{n_{tg} \times d}$.

Next, we execute GNNs-based algorithm on the RAG to update the representation of relation nodes. R-GCN (Schlichtkrull et al., 2018) is chosen as the message propagation strategy here because RAG is naturally a heterogeneous graph:

$$\bar{\mathbf{V}}_{tp}, \bar{\mathbf{V}}_{tg}, \bar{\mathbf{V}}_{re} = \texttt{R-GCN}\left(\tilde{\mathbf{V}}_{tp}, \tilde{\mathbf{V}}_{tg}, \mathbf{V}_{re}\right). \tag{6}$$

After that, for the type enhancing branch, we first calculate the *mapping probability* of an instance to each relation based on the type probability $\mathbf{p}_{tp}$ and corresponding bipartite rationale mapping $\mathbf{M}_{tp2re} \in \mathbb{R}^{n_{tp} \times n_{re}}$ (i.e., the edge weight $\mathcal{M}_{tp2re}$), and then weight the updated relation embeddings based on the mapping probability to obtain type enhancing vector $\mathbf{h}_{tp} \in \mathbb{R}^{d}$. Meanwhile, similar operations are performed in the trigger branch:

$$\begin{aligned} \mathbf{h}_{tp} &= \mathbf{p}_{tp} \mathbf{M}_{tp2re} \bar{\mathbf{V}}_{re}, \\ \mathbf{h}_{tg} &= \mathbf{p}_{tg} \mathbf{M}_{tg2re} \bar{\mathbf{V}}_{re}. \end{aligned} \tag{7}$$

4.3 Classification Module

The output module combines the overall vector and two enhancing features to get final representation, which is fed into a multi-layer perceptron followed by a softmax function for relation classification:

$$\begin{aligned} \mathbf{h}_{re} &= [\mathbf{h}_{cls}; \mathbf{h}_{tp}; \mathbf{h}_{tg}], \\ \mathbf{p}_{re} &= \texttt{SoftMax}\left(\texttt{MLP}_{\{3d,n_{re}\}}(\mathbf{h}_{re})\right). \end{aligned} \tag{8}$$

4.4 Training Objectives

Recall that there are totally three tasks in our model, including relation extraction, type prediction, and trigger (start and end indexes) labeling, which are all reduced to the classification problem. In optimization, we train the model end-to-end in a multi-task manner here, and adopt cross-entropy as the loss function for each task:

$$\mathcal{L}_{task} = \texttt{CrossEntropy}(\mathbf{y}_{task}, \mathbf{p}_{task}), \tag{9}$$

where $\mathbf{y}_{task}$ denotes the ground truth, represented by one-hot vector, $\mathbf{p}_{task \in \{re,tp,sta,end\}}$ is the estimated probability for each class.

Towards learning to perceive the strong signal that a known trigger exactly in the text, we utilize contrastive loss (Hadsell et al., 2006). The intuition is that the trigger in text h_{pre}^{tg} and the matched trigger in RAG v_{tg}^{mat} should have similar representations (i.e., have a small distance in vector space, d). For the mismatched trigger, we expect a margin m between their embeddings. The contrastive loss of trigger matching is as follows, where $\mathbb{1}_{mat}$ is 1 if a trigger is originally in the text and 0 if it is not:

$$\begin{aligned} d &= ||\mathbf{h}_{pre}^{tg} - \mathbf{v}_{tg}^{mat}||_2, \\ \mathcal{L}_{mat} &= (1 - \mathbb{1}_{mat})(\max\{0, m - d\})^2 \\ &\quad + (\mathbb{1}_{mat})(d)^2 . \end{aligned} \tag{10}$$

The joint loss of trigger labeling is thus

$$\mathcal{L}_{tg} = \mathcal{L}_{sta} + \mathcal{L}_{end} + \mathcal{L}_{mat}. \tag{11}$$

Finally, the losses from the main RE task and two subtasks are aggregated to form the training objective, with two weight factors λ_{tp} and λ_{tg}:

$$\mathcal{L} = \mathcal{L}_{re} + \lambda_{tp}\mathcal{L}_{tp} + \lambda_{tg}\mathcal{L}_{tg}. \tag{12}$$

Extension. Here, we introduce a simple extension to simultaneously make full use of all data with relation label and any number of data with pattern annotation. Specifically, when there are intact pattern annotations for an instance, we set $\mathbb{1}_{ext}$ to 1 and calculate the losses of type prediction and trigger labeling. Otherwise, we do not calculate them and set $\mathbb{1}_{ext}$ to 0. In this way, the training objective (Equation 12) is modified as follow,

$$\mathcal{L}^{'} = \mathcal{L}_{re} + \mathbb{1}_{ext}(\lambda_{tp}\mathcal{L}_{tp} + \lambda_{tg}\mathcal{L}_{tg}). \tag{13}$$

5 Experiments

We name our proposed model RARE[3], which can be adopted to both document-level and sentence-level RE tasks. Due to the differences in data formats, applicable baseline models, and the custom

[3] abbreviation of RAtionale enhanced Relation Extraction

Model	Dev		Test	
	F1 $\pm \sigma$	F1c $\pm \sigma$	F1 $\pm \sigma$	F1c $\pm \sigma$
Majority (Yu et al., 2020)	38.9 $\pm$ 0.0	38.7 $\pm$ 0.0	35.8 $\pm$ 0.0	35.8 $\pm$ 0.0
CNN (Yu et al., 2020)	46.1 $\pm$ 0.7	43.7 $\pm$ 0.5	48.0 $\pm$ 1.5	45.0 $\pm$ 1.4
LSTM (Yu et al., 2020)	46.7 $\pm$ 1.1	44.2 $\pm$ 0.8	47.4 $\pm$ 0.6	44.9 $\pm$ 0.7
BiLSTM (Yu et al., 2020)	48.1 $\pm$ 1.0	44.3 $\pm$ 1.3	48.6 $\pm$ 1.0	45.0 $\pm$ 1.3
BERT (Devlin et al., 2019)	60.6 $\pm$ 1.2	55.4 $\pm$ 0.9	58.5 $\pm$ 2.0	53.2 $\pm$ 1.6
TypeKD$_{BERT}$ (Zhang et al., 2020)†	62.4 $\pm$ 1.1	57.7 $\pm$ 1.0	60.8 $\pm$ 1.5	55.6 $\pm$ 1.4
RARE$_{BERT}$ (ours)	**64.6** $\pm$ 0.7	**60.1** $\pm$ 0.8	**64.2** $\pm$ 1.2	**58.7** $\pm$ 1.1
BERTs (Yu et al., 2020)	63.0 $\pm$ 1.5	57.3 $\pm$ 1.2	61.2 $\pm$ 0.9	55.4 $\pm$ 0.9
TypeKD$_{BERTs}$ (Zhang et al., 2020)†	65.1 $\pm$ 1.2	59.4 $\pm$ 0.9	63.5 $\pm$ 1.3	57.8 $\pm$ 1.2
RARE$_{BERTs}$ (ours)	**67.5** $\pm$ 0.8	**62.6** $\pm$ 1.0	**66.4** $\pm$ 0.8	**61.0** $\pm$ 1.0

Table 1: Main results on the document-level RE (DialogRE) task, σ denotes the standard deviation computed from five independent runs of each model. † marks the results we reproduce based on the official released code.

in handling entities, we conduct two sets of experiments, comparing RARE to their respective state-of-the-art models on the two tasks. In the experiment, we take *bert-base-uncased* as backbone encoder to verify the effectiveness of RARE and perform further analysis. Besides, we reproduce TypeKD (Zhang et al., 2020) as an extra baseline, which is a recent work using global statistics between entity types and relations in RE.

Implementation Details. We follow the same input format and hyper-parameter settings as in baselines for fair composition. Besides, the layer number of RAG is set to 2 (chosen from $\{1, 2, 3\}$), the match margin in $\mathcal{L}_{mat}$ is set to 0.1 (chosen from $\{1, 0.1, 0.01\}$) for the two sets of experiments. We tune the loss weights λ_{tp} and λ_{tg} with grid search (chosen from $[0.01, 0.05]$ in steps of 0.01) and set λ_{tp} to 0.01 and λ_{tg} to 0.03. For the nodes in RAG, we regard entity types, triggers, and relations as plain text, then employ the encoding module to achieve their initial embeddings. All the hyper-parameters are tuned based on dev set.

Evaluation Metrics. Following popular choices and previous work, we use F1/F1c scores as evaluation metrics in the document-level RE task (i.e., DialogRE), where F1c is computed by only taking in the early part of a dialogue as input, instead of the entire dialogue. In the sentence-level RE task (i.e., TACRED/V), we report micro-averaged Precision, Recall, and F1 scores.

5.1 Document-Level Relation Extraction

DialogRE (Yu et al., 2020) is a human-annotated document-level RE dataset constructed from the transcripts of an American television situation comedy *Friends*. It is also the first RE dataset with both entity type and trigger annotation.

	P	R	F1
Type Prediction	79.3	77.4	78.3
Trigger Labeling	51.5	54.2	52.7

Table 2: Performance of two subtasks on DialogRE.

5.1.1 Experimental Setup

We employ BERT and BERTs (Yu et al., 2020) as the encoding module of RARE in this task. BERTs is a speaker-aware version of BERT, achieving the best performance on the dataset. For the completeness of experiments, we include all official baselines: Majority strategy and CNN/LSTM/BiLSTM-based models (Yu et al., 2020).

5.1.2 Results and Analysis

Main Results. Comparing the performance of different models in Table 1, the first conclusion we draw is that RARE$_{BERTs}$ outperforms all baseline models in all evaluation matrices, which demonstrates the effectiveness of our rationale enhanced approach, as well as the motivation of using global pattern co-occurrence statistics to boost the performance of RE models. Secondly, RARE$_{BERTs}$ improves by a relative margin against RARE$_{BERT}$. It is strong evidence that RARE is flexible enough to adapt to various encoders. Thus, we have reason to believe that a more powerful encoding module could bring further performance gain for RARE. Lastly, TypeKD-based models have a similar trend, but their performance is relatively worse than models based on RARE, which shows that trigger and type are two non-overlapping information sources, and only considering one of them is not enough to capture complete indicative knowledge.

We report the performance of RARE$_{BERTs}$ on the two subtasks in Table 2. From the results, we find

	Dev F1
RARE$_{BERT}$	64.6
w/o Rationale graph	62.3
w/o Type enhancing branch	62.8
w/o Trigger enhancing branch	63.3
w/o Trigger matching loss	64.0
w/o Probabilistic edge weights	63.7
w/o Gate mechanism & GNNs	63.5

Table 3: Ablation study on DialogRE dev set.

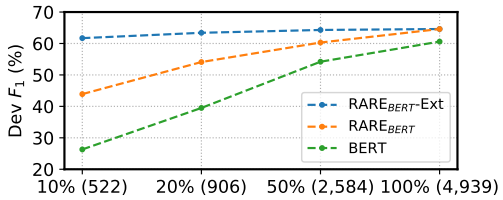

Figure 4: Performance of models on DialogRE dev set with partial training data. The positive instance number with pattern annotation is shown in brackets.

that type prediction is relatively simpler than trigger labeling. We explain that the entity type is a kind of shallow linguistic feature, while the labeling trigger requires a full understanding of context semantics. We also notice that trigger labeling performance is even worse than that of RE, since about half of the positive instances have no explicit trigger (Yu et al., 2020), meaning that the recognition of trigger faces a more serious data imbalance problem than RE. Overall, there is still a long way to improve the performance of these two subtasks, which can be left as a possible future direction.

Ablation Study. To investigate the effectiveness of each module in RARE, we conduct an ablation study on the DialogRE dev set. From the ablations in Table 3, we observe that: (1) Rationale graph is a necessary component that contributes 2.3% F1. The performance superiority of this ablation over BERT also shows that the two auxiliary subtasks of type prediction and trigger labeling are beneficial to RE. (2) Without the type or trigger enhancing branch, the performance degradation suggests that both type and trigger are necessary for our RARE. (3) The ablation of removing the trigger matching loss hurts the final result by 0.6% F1, which justifies the design philosophy of entrusting the model with the ability to perceive whether the trigger is exactly in text. (4) We also try to remove the probabilistic edge weights in RAG to make it degenerate into a standard heterogeneous graph. In that case, the performance drops by 0.9% F1. We think that such probabilistic weights are capable of carrying more global information than one-hot constraints. (5) The information propagation (i.e., gate mechanism and GNNs) brings the improvement of 1.1% F1, which provides a channel to integrate the features of input instance in the output layer.

Labor-Efficiency Study. Considering that most RE datasets have no trigger annotation, we seek to study the cost-effectiveness of adding patterns as additional annotation in this experiment. Accordingly, we explore the performance of RARE$_{BERT}$ and BERT for various fractions of training data. From Figure 4, we can see that RARE$_{BERT}$ with pattern annotations delivers competitive or even better performance as BERT with twice the traditional training data. The drastic performance gain justifies the slightly additional cost incurred in annotating patterns. Furthermore, we also introduce RARE-Ext, the extension of RARE, to fully use the partial data with pattern annotations and the remaining data with only relation labels in training, which provides a plug-and-play manner to utilize pattern annotations. The results show that with the increase of annotations, the performance improvement becomes less significant. When using 50% (with 2,584 positive instances) pattern annotations, the performance of the model is comparable to that of 100% annotations.

5.2 Sentence-Level Relation Extraction

In this section, we evaluate RARE on the sentence-level RE task with two datasets TACRED (Zhang et al., 2017) and TACREV (Alt et al., 2020). TACRED is the most widely used sentence-level RE dataset that constructed from *New York Times*. The recent TACREV (a.k.a TACRED-Revised) dataset has the same training set as TACRED, which corrects the wrong labels in the dev and test sets.

5.2.1 Experimental Setup

To our knowledge, SpanBERT (Joshi et al., 2020) is the best performance model without external knowledge in TACRED. We employ it as another encoder (besides BERT) for RARE. For completeness, we also include two official baselines, LSTM and PA-LSTM (Zhang et al., 2017), as well as two recent graph-based models, AG-GCN (Guo et al., 2019) and LST-AGCN (Sun et al., 2020), here.

Different from DialogRE, TACRED/V annotates only entity types. Inspired by the results of the label-efficiency study on DialogRE, we annotate

Model	TACRED			TACREV		
	P	R	F1	P	R	F1
LSTM (Zhang et al., 2017)	65.7	59.9	62.7	71.5	69.7	70.6
PA-LSTM (Zhang et al., 2017)	65.7	64.5	65.1	74.5	74.1	74.3
AG-GCN (Guo et al., 2019)	73.1	60.9	68.2	77.7	73.4	75.5
LST-AGCN (Sun et al., 2020)	-	-	68.8	-	-	-
BERT (Devlin et al., 2019)‡	67.2	69.3	68.2	76.0	75.6	75.1
TypeKD$_{BERT}$ (Zhang et al., 2020)†	70.6	68.7	69.6	77.9	76.1	77.0
RARE$_{BERT}$-Ext (ours)⋆	71.4	68.1	**69.8**	78.6	76.2	**77.4**
SpanBERT (Joshi et al., 2020)	70.8	70.9	70.8	75.7	80.7	78.0
TypeKD$_{SpanBERT}$ (Zhang et al., 2020)†	71.7	70.4	**71.0**	79.8	78.3	78.8
RARE$_{SpanBERT}$-Ext (ours)⋆	72.5	69.3	70.8	80.1	78.0	**79.0**

Table 4: Main results on the sentence-level RE (TACRED/V) task. ‡ marks the results we reproduce based on the repository released by (Joshi et al., 2020). We implement RARE-Ext with 20% extra annotations (⋆).

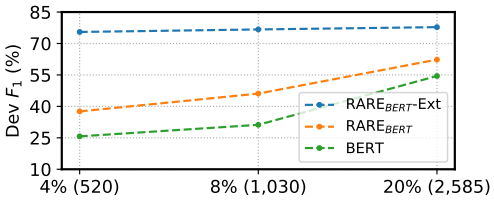

Figure 5: Performance of models on TACREV dev set with partial training data.

triggers for 2,585 positive instances, which accounts for about 20% of all positive instances in the training set of TACRED/V, to verify whether RARE could maintain such excellent lab efficient performance on sentence-level RE task. We repeat our experiments for five random seed initializations, and the results are statistically significant with a p-value of less than 0.05.

5.2.2 Results and Analysis

Main Results. With 20% pattern annotations, we compare RARE-Ext against several representative baselines and summarize the results in Table 4. Similar observations hold that RARE is capable of achieving superior performances with advanced encoding modules. Moreover, RARE-Ext achieves or even surpasses the performance of TypeKD that using 100% type annotations. Although sometimes RARE does not make significant improvements on TACRED, it outperforms the baselines in TACREV and leads to state-of-the-art performances, which is a more accurate evaluation set. Overall, the performance gain of RARE on this task is not as amazing as the document-level task. We analyze that because the sentence is much shorter than the document, and involves fewer relations, BERT-based models are sufficient in capturing the key semantic clue for decision-making, thus the benefits of global knowledge are slightly limited.

Labor-Efficiency Study. Following the approximate number of positive instances in DialogRE, we split the pattern-annotated data to perform the labor-efficiency study on TACREV (see Figure 5). The results indicate that when both using partial data, RARE$_{BERT}$ consistently outperforms BERT. It enlightens us to fully exploit the potential knowledge of the dataset, including local annotation and global statistics, to improve the performance of RE, especially under a low-resource scenario. The considerable progress of RARE$_{BERT}$-Ext demonstrates that RARE is able to improve RE by annotating patterns on any part of an existing dataset. Considering the differences between DialogRE and TACREV (e.g., relation number, domain and style, the ratio of positive and negative instances), it is under investigation whether further improvements could be made by increasing annotations on TACREV, and we leave it as future work.

5.3 Case Study

In Figure 6, we select two representative cases to demonstrate the working principle of RARE. The first case is a short snippet from a DialogRE document, in which two entities are scattered in different sentences, and the context semantics is complex and changeable, BERT fails to capture the relation between them. Conversely, RARE predicts the trigger `engaged` and aligns it with the known trigger `engagement`, and then highlights the strong signal to identify the relation correctly. In the second case, which is from TACREV, BERT mistakenly regards `Jackson Hewitt` as a person, leading to a wrong answer of person-related relation. With the help of type prediction and the global type-relation

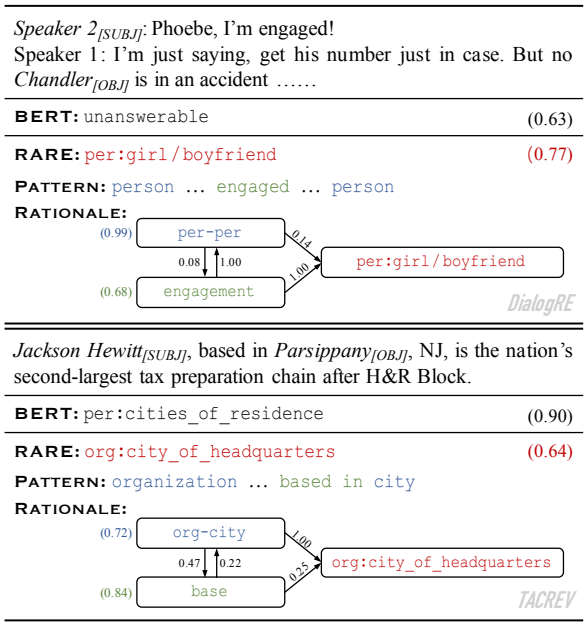

Figure 6: Internal principles of RARE. The number in bracket refers the probabilities predicted by model.

constraints in RAG, RARE could avoid this error and make the right decision.

6 Conclusion

In this paper, we propose a novel rationale graph to organize the global co-occurrence statistics among entity types, triggers, and relations. By introducing the two subtasks of entity type prediction and trigger labeling, we build the connection between input instance and the known patterns in rationale graph, which provides the model with the possibility to benefit from the global co-occurrence knowledge stored in the graph, so as to improve the performance of RE. Experimental results on two public datasets prove the effectiveness of our method. We also highlight two directions for future work: the first is to improve the performance of two subtasks, especially trigger labeling, the other is to adopt the proposed approach in more RE scenarios.

Acknowledgments

We would like to thank all reviewers for their insightful comments and suggestions. We would also like to thank Ning Cong, Minghuan Yuan, Gehang Zhang, and Haoran Xing for their help. This work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences (grant No. XDC02040400), and the Youth Innovation Promotion Association of Chinese Academy of Sciences (grant No.2021153).

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. Tacred revisited: A thorough evaluation of the tacred relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1558–1569.

Chia-Hui Chang and Shao-Chen Lui. 2001. Iepad: Information extraction based on pattern discovery. In *Proceedings of the 10th International Conference on World Wide Web (WWW)*, pages 681–688.

Nick Chater, Joshua B Tenenbaum, and Alan Yuille. 2006. Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, 10(7):287–291.

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 Technologie (NAACL-HLT)*, pages 4171–4186.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 241–251.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1735–1742.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip S Yu. 2020. Selfore: Self-supervised relational feature learning for open relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3673–3682.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics (TACL)*, 8:64–77.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 178–181.

Jun Kuang, Yixin Cao, Jianbing Zheng, Xiangnan He, Ming Gao, and Aoying Zhou. 2020. Improving neural relation extraction with implicit mutual relations. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1021–1032.

Bill Yuchen Lin, Dong-Ho Lee, Ming Shen, Ryan Moreno, Xiao Huang, Prashant Shiralkar, and Xiang

Ren. 2020. Triggerner: Learning with entity triggers as explanations for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8503–8511.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2124–2133.

Yang Liu, Kang Liu, Liheng Xu, and Jun Zhao. 2014. Exploring fine-grained entity type constraints for distantly supervised relation extraction. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 2107–2116.

R Mooney. 1999. Relational learning of pattern-match rules for information extraction. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI)*, pages 328–334.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from context or names? an empirical study on neural relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of the 2018 European Semantic Web Conference (ESWC)*, pages 593–607.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2895–2905.

Yu Su, Honglei Liu, Semih Yavuz, Izzeddin Gür, Huan Sun, and Xifeng Yan. 2018. Global relation embedding for relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 820–830.

Kai Sun, Richong Zhang, Yongyi Mao, Samuel Mensah, and Xudong Liu. 2020. Relation extraction with convolutional network over learnable syntax-transport graph. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 8928–8935.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. Reside: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1257–1266.

Zhen Wang, Jennifer Lee, Simon Lin, and Huan Sun. 2020. Rationalizing medical relation prediction from corpus-level statistics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8078–8092.

Bowen Yu, Zhenyu Zhang, Tingwen Liu, Bin Wang, Sujian Li, and Quangang Li. 2019. Beyond word attention: using segment attention in neural relation extraction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5401–5407.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4927–4940.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 2335–2344.

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 (EMNLP)*, pages 2205–2215.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 35–45.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1441–1451.

Zhenyu Zhang, Xiaobo Shu, Bowen Yu, Tingwen Liu, Jiapeng Zhao, Quangang Li, and Li Guo. 2020. Distilling knowledge from well-informed soft labels for neural relation extraction. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 9620–9627.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 427–434.