BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 10/10 pp248–262
Volume 6, Number 2, June 2023
DOI: 10.26599/BDMA.2022.9020051

Denoising Graph Inference Network for Document-Level Relation Extraction

Hailin Wang, Ke Qin, Guiduo Duan*, and Guangchun Luo

Abstract: Relation Extraction (RE) is to obtain a predefined relation type of two entities mentioned in a piece of text, e.g., a sentence-level or a document-level text. Most existing studies suffer from the noise in the text, and necessary pruning is of great importance. The conventional sentence-level RE task addresses this issue by a denoising method using the shortest dependency path to build a long-range semantic dependency between entity pairs. However, this kind of denoising method is scarce in document-level RE. In this work, we explicitly model a denoised document-level graph based on linguistic knowledge to capture various long-range semantic dependencies among entities. We first formalize a Syntactic Dependency Tree forest (SDT-forest) by introducing the syntax and discourse dependency relation. Then, the Steiner tree algorithm extracts a mention-level denoised graph, Steiner Graph (SG), removing linguistically irrelevant words from the SDT-forest. We then devise a slide residual attention to highlight word-level evidence on text and SG. Finally, the classification is established on the SG to infer the relations of entity pairs. We conduct extensive experiments on three public datasets. The results evidence that our method is beneficial to establish long-range semantic dependency and can improve the classification performance with longer texts.

Key words: Relation Eextraction (RE); document-level; denoising; linguistic knowledge; attention mechanism

1 Introduction

The document-level Relation Extraction (RE), which aims to detect a relationship between entity mentions from raw text, plays a critical role in addressing the issue of information extraction. Conventional works that obtained relational facts within a single sentence (sentence-level) ignored these complex facts across multiple sentences. Over the past few years, researches on the document-level RE[1–7] provide in-depth insights into the RE task, where transformer-based and graph-based methods are widely applied. All these methods suffer from noise in the text, and a necessary long-range semantic dependency among all mentions is a way around this issue. Nevertheless, too little work is devoted to constructing a document's long-range semantic dependency between entity pairs.

Conventional sentence-level methods[8–10] using the Syntax Dependency Tree (SDT) give some insights into coping with this problem in the document-level RE. The sentence-level RE task exploits the SDT to graphically present word intradependency relation. The critical point is that the trimmed SDT, the Shortest Dependency Path (SDP), cuts the sentence to a short sequence, thus guiding the model to extract relations between entity pairs. This SDP method builds the long-range semantic dependency between entities by transforming the sentence-level RE task into a graph

• Hailin Wang is with School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu 611130, China. E-mail: wanghl@swufe.edu.cn.

• Ke Qin, Guiduo Duan, and Guangchun Luo are with School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. E-mail: qinke@uestc.edu.cn, guiduo.duan@uestc.edu.cn; gcluo@uestc.edu.cn.

* To whom correspondence should be addressed.

Manuscript received: 2022-10-07; revised: 2022-12-05; accepted: 2022-12-12

denoising paradigm. This paradigm first constructs a graph and then extracts a subgraph by eliminating nodes to retain a piece of keyword set, which works well for the sentence-level RE, and may apply to the document-level RE. Following the sentence-level paradigm, two steps are necessary to transform this denoising paradigm for the document-level task. The first step is to represent intradependency and interdependency relations among words and sentences for a document, and we introduce discourse dependency relation to illustrate the interdependency relation. The second step is to remove irrelevant words graphically to achieve the purpose of denoising. To show the feasibility of this denoising paradigm for document-level RE task, Fig. 1 illustrates a long text example[11] of building the long-range semantic dependency by denoising. In Fig. 1, a word set organizes the simple graph as a trimmed sentence "Julian Reinard is German footballer appeared in German Bundesliga" by collecting all entities and their semantic dependency words along the syntax and discourse dependency relation. Inferencing this word set reasonably makes it easy to deduce the relationship among the three entities (Julian Reinard, German, Bundesliga). This simple graph shows that carefully picking out some keywords relevant to those entities in a long text is enough to infer all relationships among entities. In other words, graphically constructing a long-range semantic dependency using a denoising method could benefit the document-level RE. Based on this observation, transforming the sentence-level denoising paradigm to a document-level RE should figure out the following three problems: (1) create a document-level graph using various linguistic knowledge, e.g., syntax and discourse dependency relation, (2) denoise the graph carefully as the SDP does, and (3) infer all relations on the denoised graph.

In this work, we propose a Denoised Graph Inference model (DGI) to address the abovementioned issues. We construct a basic document-level graph, develop a novel graph-based denoising method for RE, and utilize a slide residual attention mechanism to reason entity pairs' relationship on the denoised graph. Figure 2 shows the whole model architecture.

First, this paper introduces the Rhetorical Structure Theory (RST)[12], as an external linguistic knowledge to analyze the discourse association among multiple Elementary Discourse Units (EDUs). Furthermore, a graph constructor combines the syntax and discourse dependency relation from SDT and RST to construct a basic graph SDT-forest.

Second, our mention-level denoised graph SG is constructed from the abovementioned SDT-forest. A conventional SDP with two entities can easily estimate the shortest path between two entities using a simple algorithm. Nevertheless, for the document-level RE with more than three mentions, it is not easy to describe the minimal requirement of the semantic keywords. The Steiner tree algorithm can generate a minimum-spanning

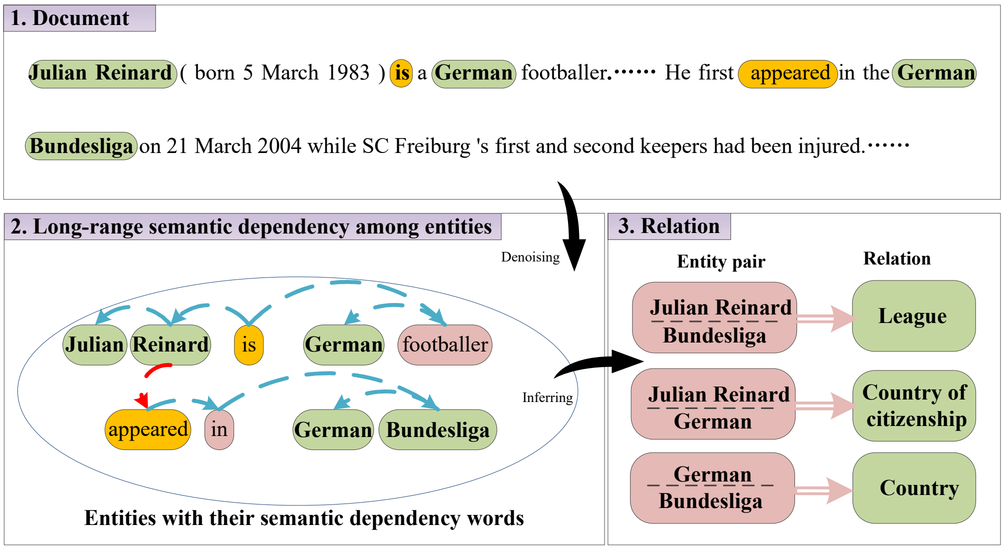

1. Document
Julian Reinard (born 5 March 1983) is a German footballer.······ He first appeared in the German
Bundesliga on 21 March 2004 while SC Freiburg 's first and second keepers had been injured.······
2. Long-range semantic dependency among entities
Denoising
Julian
Reinard
is
German
footballer
appeared
in
German
Bundesliga
Inferring
Entities with their semantic dependency words
3. Relation
Entity pair
Relation
Julian Reinard
Bundesliga
League
Julian Reinard
German
Country of citizenship
German
Bundesliga
Country

Fig. 1 Example of discovering a long-range semantic dependency among entities in a graph manner. Figure 1 presents three cards, in which Card 1 shows a document instance and its entities, Card 2 picks up minimal evidence (subgraph) along the syntax and discourse relation (blue or red line) to build the long-range semantic dependency for various entities (i.e., Julian Reinard, German, and Bundesliga), and Card 3 indicates the relation between each entity pair.

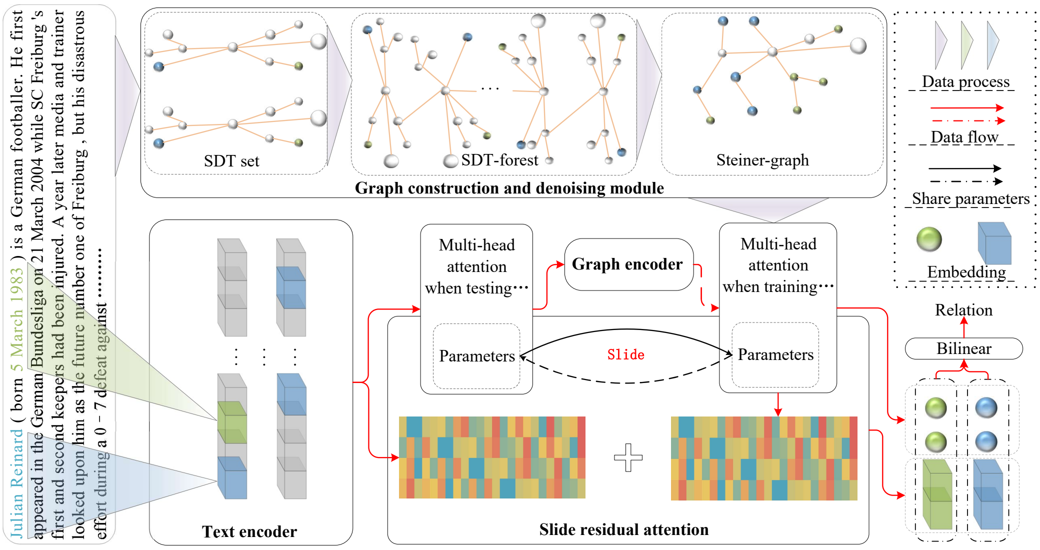

Julian Reinard (born 5 March 1983) is a German footballer. He first appeared in the German Bundesliga on 21 March 2004 while SC Freiburg's first and second keepers had been injured. A year later media and trainer looked upon him as the future number one of Freiburg , but his disastrous effort during a 0 - 7 defeat against ……
SDT set
SDT-forest
Steiner-graph
Graph construction and denoising module
Data process
Data flow
Share parameters
Embedding
Multi-head attention when testing…
Graph encoder
Multi-head attention when training…
Relation
Bilinear
Parameters
Slide
Parameters
Text encoder
Slide residual attention

Fig. 2 Framework of DGI model.

tree that contains all required terminals and additional vertices. We use the Steiner tree algorithm with all mentions as the terminals to extract a subgraph from the SDT and assume the subgraph as the long-range semantic dependency for all mentions.

Finally, after obtaining the mention-level denoised graph, the Densely Connected Graph Convolutional Network (DCGCN)[13] is deployed to capture a word-level feature. For performance improvement, our slide residual attention highlights two different word attention weights to enhance the contextual presentation and infer the classification using the denoised graph feature.

The main contributions of our work are summarized as follows:

(1) This article defines a simple document-level graphically denoising method, which utilizes syntactic and discourse relations to present a document in a novel graphical way and then eliminates the linguistically irrelevant words to build a long-range semantic dependency among entity mentions.

(2) We propose a slide residual attention mechanism that dynamically chooses two levels of word features from the original text and the denoised graph as complementary elements to infer the corresponding relation of entity pairs.

(3) We conduct extensive experiments on three standard datasets and present detailed analyses. Our experiment shows that the denoising method has a better effect and can bring better performance for longer text instances.

2 Related Work

Early sentence-level RE builds long-range semantic dependency using the SDP methods[14–17] by denoising words of low linguistic relevance. The document-level RE must infer from multiple sentences, resulting in the sentence-level denoising methods being unable to cope with a document. However, there are still many other approaches for the document-level RE, including transformer- and graph-based methods.

Transformer-based methods deal with a document in a sequence-to-sequence manner. Wang et al.[1] used a two-step strategy to divide the task into two subtasks, yielding a better result. Han and Wang[3] inserted some special tokens to identify all mention positions and types to enhance the entity representation. Tang et al.[2] constructed a hierarchical inference network, reasoning at the entity, sentence, and document levels to output the final prediction. Zhou et al.[18] proposed a transformer-based model of localized context pooling technique and adaptive-thresholding. Eberts and Ulges[4] jointly predicted the entity, entity type, and relation step by step. Yuan et al.[6] designed a document using two-level features to predict relations. Xue et al.[19] used simple multiple [CLS] tokens with a relation refinement gate to capture possible relations between different pairs of entities mentioned in the dialogue. These studies focused only on the local entity representation, ignoring the global links among words or phrases in a document.

Graph-based methods effectively provide a reasoning ability considering more association across words in a flattering text. Nan et al.[20] refined a latent document-level graph with mention, entity, and meta-dependency to reason the relation from the document. Zeng et al.[21] generated an efficient inference path for the RE from a mention-level graph showing all entities globally. Li et al.[22] developed a graph-enhanced dual attention network to characterize the interaction between sentences and relation instances. Christopoulou et al.[23] constructed a heterogeneous graph with three types of nodes and edges, iteratively modeling the long-range semantic dependency among entity pairs over a document. Xue et al.[24] generated a latent multi-view graph using a Gaussian graph generator to capture the possible relationships among tokens. Li et al.[7] devised a heterogeneous affinity graph inference network with noise suppression mechanism to build the long-distance reasoning chain in document-level RE.

However, the above works rarely considered denoising a document to physically build the long-range semantic dependency between entity pairs. But some previous sentence-level approaches[9, 16, 25–28] graphically trimmed the sentence to a short sequence to obtain an excellent result. These methods effectively use the syntax dependency relation or the intradependency. Nevertheless, only a few studies[10, 29, 30] introduced the interdependency or simple discourse (inter-sentence) relation to the RE. Inspired by the RST[12], the most accepted discourse analysis framework, we consider introducing this well-organized representation of documents and utilizing their discourse-level segmentation to model inter-sentence semantic dependency with more refined granularity. This can help us to merge the intradependency and interdependency among words and sentences in a document and establish the long-range semantic dependency by denoising the text. The RST transforms a document into a DEPendency-based Discourse Tree (DEP-DT)[¶] with EDUs, which explicit pinpoints the critical interdependence relation (e.g., example, elaboration, concession, consequence, and contrast). This discourse relation has been applied in question and answering[32], summarization[33, 34], and translation[35].

Consequently, in this study, we leveraged intradependency and interdependency relationships from the SDT and the DEP-DT (shown in Fig. 3) to form a novel document-level graph for denoising like the SDP. Further, we proposed a slide residual attention to dynamically choose two levels of word features as complementary elements to infer the entity pairs' relation on a denoised document graph.

3 Method

This paper proposes a DGI model for document-level relation extraction using a denoising method to capture the long-range semantic dependency among all entities. Figure 2 shows the whole model structure. This section presents the DGI in a pipeline: the text encoder, graph construction and denoising modules, graph encoder, and slide residual attention modules.

3.1 Text encoder module

Given a document $\mathcal{D} = [S_1, S_2, \ldots, S_n]$, which contains l words and n sentences, and each sentence $S_i = [x_1, x_2, \ldots, x_m]$ has m words. In this document, two different inserted special marks ("##1" and "ORG") indicate the entity positions and types[†] at the start and end of each mention. A previous work[36] shows these composite marks' effectiveness. Following Ref. [11], the text encoder module converts the document $\mathcal{D}$ to contextual embedding as follows by using a pre-trained language model:

$$H_e = [h_1, h_2, \ldots, h_l] = Encoder(\mathcal{D}) \tag{1}$$

Generally, the pre-trained language model encodes sentences less than 512 words. However, after inserting the composite marks and tokenization, the longest sentence would be more than 1024 words. Hence, we concatenate two encoder modules to obtain more complete sentence embedding sequences. Meanwhile, the final sentence would discard any word longer than 1024 tokens. This work follows Ref. [37], applying logsumexp pool to obtain entity embedding h_{e_i}. Each embedding corresponds to the first start marker "##1" of mentions. In the document $\mathcal{D}$, any entity e_i has m mentions. Thus, our DGI denotes each entity as follows:

$$h_{e_i} = logsumexp(h_1, h_2, \ldots, h_m) = \log \sum_{j=1}^{m} \exp(h_j) \tag{2}$$

3.2 Graph construction and denoising module

We use the linguistics knowledge SDT and DEP-DT as a tool to transform document $\mathcal{D}$ to a graph and get a subgraph. To this end, we parse a document to EDU pieces and further parse each sentence of the document to the SDT, forming an SDT tree set. A designed

[¶] The transformation algorithm of DEP-DT follows Ref. [31].

[†] These marks contain "##1", "##2", and so on. Type marks convert type words into abbreviations, such as "blank" to "BLANK" and "organization" to "ORG".

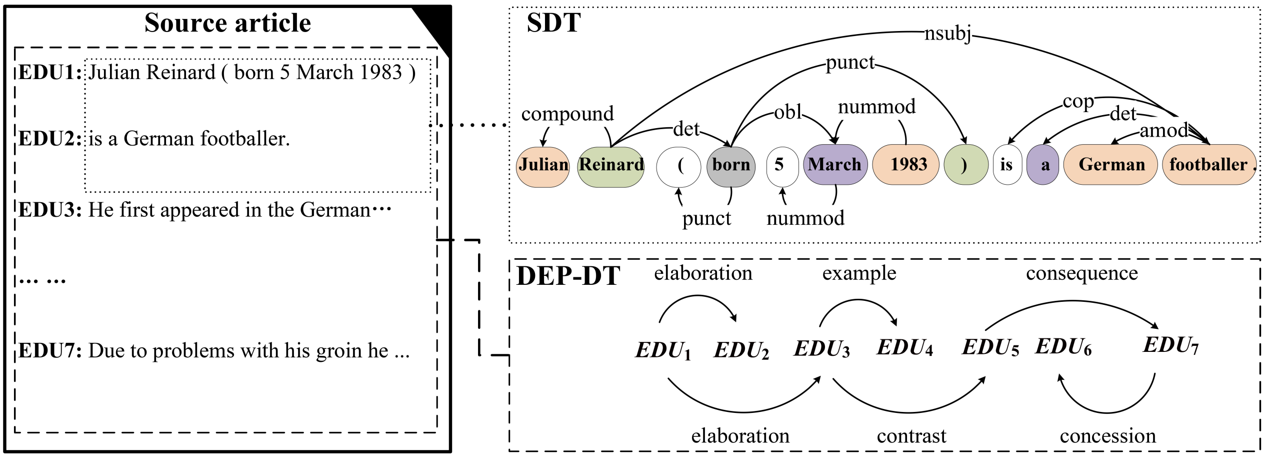


Fig. 3 Example of SDT and DET-DT, which are parsed from a sentence or the whole document. For any document, each sentence has an SDT, but just with one DEP-DT.

algorithm connects each EDU through links in the SDT and DEP-DT, yielding a word-level graph, that is, SDT-forest. However, this simple graph is not our target. We want an acceptable denoising method for the document-level RE, like the SDP in the sentence-level RE. We exploit the Steiner tree as an alternative method to find the minimum spanning-tree over a graph. Hence, the following subsection includes three parts: syntax and discourse dependency parsing, graph connection with linguistic knowledge, and denoise with the Steiner tree.

3.2.1 Syntax and discourse dependency parsing

The SDT and DEP-DT research has a long tradition. In this work, intradependency and interdependency relations across words in a document depend on these two. As a subset sequence of the SDT, the SDP plays a critical role in the sentence-level RE field. Despite its prolonged success, the SDP is limited in sentence-level RE. Syntax dependency only explains the relationship between the words within a sentence without the relationship between the words across sentences. Hence, it is not suitable for the document-level RE. The discourse dependency, which fills this gap somewhat, has exerted preliminary efforts to interpret the relationship between the words across sentences, which could benefit the document-level graph construction.

To construct the document-level graph, we first use spaCy[§] to acquire each SDT of sentences in a document. The spaCy outputs a set, including n SDTs,

$$SDT_{set} = \{t_1, t_2, \ldots, t_n\} = SpaCy([S_1, S_2, \ldots, S_n]) \tag{3}$$

For each SDT $t = \mathcal{G}(\mathcal{N}_g, \mathcal{L}_g)$, $\mathcal{N}_g$ and $\mathcal{L}_g$ correspond to the words and syntax dependency relations, respectively.

The discourse dependency parsing phrase uses the code from the DPLP[38], which parses the document $\mathcal{D}$ into a DEP-DT with segmented text EDUs. Any SDT usually includes d EDUs ($d \geqslant n$),

$$T = DPLP([S_1, S_2, \ldots, S_n]) \tag{4}$$

For any DEP-DT tree $T = \mathcal{G}(\mathcal{N}_d, \mathcal{L}_d)$, $\mathcal{N}_d$ and $\mathcal{L}_d$ correspond to $EDUs = [edu_1, edu_2, \ldots, edu_p]$ and discourse dependency relation. Each EDU includes a short word sequence, and Fig. 4 shows the visualization of the relationship between syntax and discourse dependencies.

3.2.2 Graph connection with linguistic knowledge

The intradependency and interdependency relations from t and T are exploited to link words in a document to construct the document-level graph. Actually, after getting t and T two different trees, we use three link types to construct the graph: syntax dependency relation $\mathcal{L}_g$, discourse dependency relation $\mathcal{L}_d$, and sentence adjacent relation $\mathcal{L}_a$. The **syntax dependency relation** indicates the relationship between the words in a single sentence. The **discourse dependency relation** reveals the discourse rhetoric relationship between EDUs, showing the relative importance of different context units. Lastly, the **sentence adjacent relation** shows a neighbourhood relationship within sentences.

We use an anchor node (i.e., a special word in EDU) from a subtree to present the discourse relation. The subtree composes of the intersection of nodes $\mathcal{N}_{st}$ in the tree t and words in each EDU. The connection $\mathcal{L}_{st}$ of the nodes in the subtree is the same as that of tree t. Consider a t with a word sequence $[x_q, \ldots, x_w]$ and an EDU with another word sequence $[x_y, \ldots, x_u]$,

[§] https://spacy.io

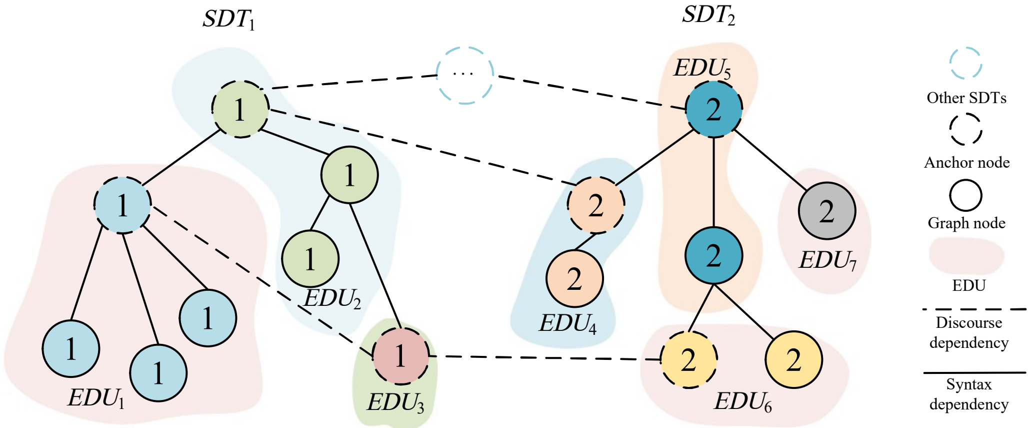


Fig. 4 Example of the SDT-forest connected by the syntax and discourse dependencies. Two SDTs are plotted and the others are omitted. The coloured areas depict the EDUs in the corresponding SDT. Each EDU has an anchor node (circle with dashed lines), which is a relative root node in the subtree (coloured area).

$\mathcal{N}_{st} = [x_q, \dots, x_w] \cap [x_y, \dots, x_u]$. Hence, a subtree is denoted as $t_i^s = \mathcal{G}(\mathcal{N}_{st}, \mathcal{L}_{st})$. The process of obtaining one anchor node (Figure 4 circle with dashed lines) is $root(t_i^s)$. All the root node set or the anchor node set is

$$\mathcal{N}_{anchor} = \left\{root(t_1^s), root(t_2^s), \dots, root(t_q^s)\right\} \quad (5)$$

With the three kinds of relationships, anchor node-set, and other words in the document, a new graph SDT-forest (shown in Fig. 4) could be denoted as $F = \mathcal{G}(\mathcal{N}, \mathcal{L})$, where $\mathcal{N} = \mathcal{D} = [x_1, x_2, \dots, x_m]$ and $\mathcal{L} = \mathcal{L}_g \cup \mathcal{L}_d \cup \mathcal{L}_a$. The relationships in $\mathcal{L}_d$ connect each corresponding node in SDT according to $\mathcal{N}_{anchor}$ instead of EDUs.

3.2.3 Denoise with Steiner tree

In the sentence-level task, the SDP is an efficient method of obtaining the keywords in a sentence. It could obtain a path along with two entity words. However, the SDP is unsuitable when we obtain the abovementioned document-level graph, namely, the SDT-forest with multiple entities and more words. Hence, a novel method, called the Steiner tree, is used to acquire the minimum spanning-tree among the whole document-level graph. The Steiner tree is an NP problem, and we take an approximation algorithm from Networkx[b] to obtain the SG $S = \mathcal{G}(\mathcal{N}_s, \mathcal{L}_s) = Steiner(F, \mathcal{N}_e)$, where $\mathcal{N}_e$ is the entity node set. $\mathcal{L}_s$ is a subset of $\mathcal{L}$ calculated by the Steiner tree, and denoted as an undirected edge $\mathcal{L}_s$ with an $s \times s$ adjacency matrix $\boldsymbol{A}$.

3.3 Graph encoder module

After getting S, this module extracts all node representations $\mathcal{R}$ from their corresponding embedding of $\mathcal{N}_s$ in $\boldsymbol{H}_e$. We then exploit the DCGCN[13] for further graph processing with adjacency matrix $\boldsymbol{A}$ and node representation $\mathcal{R}$. The DCGCN computation processing is as follows:

$$\boldsymbol{H}_g = DCGCN(\mathcal{R}, \boldsymbol{A}) \quad (6)$$

where the dimension of all parameters follows the DCGCN, $\boldsymbol{H}_g = [h_1, h_2, \dots, h_s]$, s is the number of nodes corresponding to the dimension of $\boldsymbol{A}$.

3.4 Slide residual attention

The attention weight is from the pre-trained language model and a slide multi-head module herein. Considering that the denoising method may disrupt the continuity of tokens, the one-layer multi-head module is deployed to enhance the node representation and augment the input data. Meanwhile, *slide* means that the multi-head shifts at a different position during the training/testing phase, and *residual* refers to two attentions from the encoder module and the abovementioned slide multi-head module. We argue that this shift operation will avoid overfitting and help improve the model's performance in this combination of graphics and text sequences.

Precisely, the multi-head module formulated by Ref. [39] consists of multiple linear transformations and scaled dot-product attention,

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d}}\boldsymbol{V}\right) \quad (7)$$

$$MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Con(head_1, \dots, head_h)\boldsymbol{W} \quad (8)$$

where $head_i = Attention(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V)$ and $\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{H}_g$.

[b] https://networkx.org/

Next, the multi-head module *MultiHead*($\boldsymbol{H}_g$, $\boldsymbol{H}_g$, $\boldsymbol{H}_g$) outputs embedding $\boldsymbol{H}_m$ and attention weights $\boldsymbol{A}^m \in \mathrm{R}^{head \times s \times s}$. Each token in $\boldsymbol{H}_m$ is given a $head$ attention matrix. We take the attention from the "##1" mark as the mention-level attention with the $head$ matrix. The $head$ attention matrix is averaged to a vector to present one attention to all tokens. In this way, each entity has a corresponding attention vector. The attention vector from the head e_h entity and tail e_t entity are denoted as $\boldsymbol{A}_h^m$ and $\boldsymbol{A}_t^m$, respectively. When these two vectors are combined to represent all the context, they multiply with each other. At the same time, the pre-trained language model also outputs an attention vector from the head e_h entity and tail e_t entity, denoted as $\boldsymbol{A}_h^e$ and $\boldsymbol{A}_t^e$, respectively. All attention procedures are as follows:

$$\boldsymbol{\alpha} = \{\alpha_m, \alpha_e\} = average\{(\boldsymbol{A}_h^m \cdot \boldsymbol{A}_t^m), (\boldsymbol{A}_h^e \cdot \boldsymbol{A}_t^e)\} \quad (9)$$

These two attention weights α_m and α_e can highlight the word-level evidence on both text and SG in a complementary way. We use the DCGCN module to encode nodes from the pre-trained language module and extract the first token to present the whole original words. This partial token may result in information loss. To overcome these problems, our DGI shifts the multi-head module during training/testing, called slide residual attention, which may slow down the fitting process and enrich the representation of the first token of entity mention. The attention is calculated as follows:

$$c_{residual} = \boldsymbol{H}_e \alpha_e + k\boldsymbol{H}_m \alpha_m + (1-k)\boldsymbol{H}_g \alpha_m \quad (10)$$

where $\boldsymbol{H}_e$ is contextual embedding from the encoder module, $\boldsymbol{H}_m$ is the output from the multi-head module, k is 1 or 0 during training/testing, respectively, $\boldsymbol{H}_g$ is based on $\boldsymbol{H}_m$. In model training, the multi-head module is after the DCGCN module. Instead, in testing our model, the multi-head module slides ahead of DCGCN module, and $\boldsymbol{H}_g$ is from Eq. (6) should be redefined as $\boldsymbol{H}_g = DCGCN(\boldsymbol{H}_m, \boldsymbol{A})$.

Accordingly, two entities embedding from $\boldsymbol{H}_e$ and $\boldsymbol{H}_m$ (or $\boldsymbol{H}_g$) for any entity e_i are represented as $h_{e_i}^e$ and $h_{e_i}^m$ according to Eq. (2). To obtain the final entity representation $\hat{e}$, these two entities are combined by addiction $h_{e_i}^e + h_{e_i}^m$, and then concatenated with the abovementioned residual context $c_{residual}$. Therefore, each entity and its context embedding will be fed into a fully connected layer to obtain $\hat{e}$ by linear transformation,

$$\hat{e} = \tanh(\boldsymbol{W}((h_{e_i}^e + h_{e_i}^m) : c_{residual})) \quad (11)$$

where ":" is concatenation, $\boldsymbol{W} \in \mathbf{R}^{d \times 2d}$, d is the dimension of each token. Finally, for any instance in the data set, our method obtains $\hat{e}_h$ and tail $\hat{e}_t$ entity representation.

3.5 Prediction module

For any entity pair, the abovementioned module gives two embeddings ($\hat{e}_h$ and $\hat{e}_t$). Following the DocRED[11], our method uses a bilinear function with sigmoid activation to obtain the probability of predication as follows:

$$P = (r|\hat{e}_h, \hat{e}_t) = sigmoid(\hat{e}_h^{\mathrm{T}} \boldsymbol{W}_r \hat{e}_t + b_r) \quad (12)$$

where $\boldsymbol{W}_r \in \mathbf{R}^{d \times k \times d}$ and $b_r \in \mathbf{R}^k$ are the trainable weights and bias, respectively, k is the number of relation categories. Our method operates three different loss functions: the standard cross-entropy loss and the methods adopted in Refs. [18, 40], the last of which works better.

4 Experiment

4.1 Dataset

In this work, we use three different standard public datasets, DocRED[11], and CDR[41], and GDA[42]. The first is the main experimental object, and the other two datasets are auxiliary comparative experiments. **DocRED** is a prevailing general-purpose dataset, with 96 predefined relation types, consisting of 5053 instances, of which 3053 for training, 1000 for development, and others for testing. **CDR**[41] is a chemical-induced disease dataset selecting a total of 1500 articles split into three subsets: 500 each for the training, development, and testing. **GDA**[42] contains 30 192 Medline abstracts split into 29 192 articles for training and 1000 for testing. This dataset for the biomedical domain aims to predict the associations between gene and disease concepts.

4.2 Experiment settings

This section presents in detail the three types of settings. First, the method parses a document to the SDT and the DEP-DT by exploiting spaCy and DPLP, respectively. Next, our DGI adopts prevailing pre-trained language models, including BERT-base, BERT-large, RoBERTa-large, SciBERT-base, and BioBERT[43]. Our model is then optimized with AdamW using a two-layer learning rate with a linear warmup for the first 6% of the steps, followed by a linear decay to 0. We perform early stopping based on the F1-score on the development set. Our DGI adjusts all hyperparameters on the development set. Table 1 lists these parameters.

Table 1 Hyperparameters setting.

Hyperparmeter	DocRed		CDR/GDA	
	BERT	RoBERTa	SciBERT	BioBERT
Lr 1st	1×10^{-4}	1×10^{-4}	1×10^{-4}	
Lr others	3×10^{-5}	5×10^{-5}	2×10^{-5}	
Epoch	30	30	30	
Batch size	4	4	4	
GCN layer	3	2	3	
GCN drop	0.4	0.4	0.4	

Note: “Lr 1st” and “Lr others” refer to different learning rates for the GCN and other layers, respectively.

4.3 Baseline models and metric

This experiment compares our approach with some classic methods operating on DocRED, CDR, and GDA in four types: embedding-based, embedding + graph-based, Transformer-based, and Transformer + graph-based. Our experiment results are presented in Tables 2 and 3. These methods mainly adopt two metrics to evaluate the results: the macro-averaged F1-score and Ign F1 for DocRED, F1-score for CDR and GDA. Ign F1 refers to the F1-score evaluated on the dataset that excludes the relational facts occurring in the test, development, and training sets.

4.4 Result

4.4.1 Results on DocRED

Table 2 lists the related models on DocRED. From the table, our model can predict competitive results among all models, achieving a 62.96 F1-score on the test data set (test on the CodaLab score-board[✠]). Although our test F1-score is lower than some previous work[18,21,46], our output still outperforms mostly those of RoBERTa-large work and stays ahead of other models based on Transformers + graph-based methods. In short, our model effectively works in document-level RE tasks. Compared with GAIN on RoBERTa-large, our model outputs 0.84 and 0.64 of improvement on development

[✠] https://competitions.codalab.org/competitions/20717#results

Table 2 Model comparison on DocRED.

Type	Model	Development set		Test set	
		Ign F1	F1-score	Ign F1	F1-score
Embedding-based	CNN[11]	41.85	43.45	40.33	42.26
	LSTM[11]	48.44	50.68	47.71	50.07
	BiLSTM[11]	48.87	50.94	48.78	51.06
	Context-Aware[11]	48.94	50.17	48.40	50.70
Embedding+ graph-based	AGGCN-RE[44]	46.29	52.47	48.89	51.45
	LSR[20]	48.82	55.17	52.15	54.18
Transformer-based	BERT$_{base}$[1]	–	54.16	–	53.20
	BERT-Two-Step$_{base}$[1]	–	54.42	–	53.92
	HIN-BERT[2]	54.29	56.31	53.70	55.60
	DEMMT-BERT$_{base}$[3]	55.50	57.38	54.93	57.13
	JEREX$_{base}$[4]	–	–	58.44	60.40
	CoreBERT$_{base}$[45]	55.32	57.51	54.54	56.96
	BERT+ESA$_{base}$[6]	56.20	58.28	55.71	58.04
	SSAN$_{base}$[46]	57.03	59.19	55.84	58.16
	ATLOP$_{base}$[18]	59.22	61.09	59.31	61.30
	CoreBERT$_{large}$[45]	56.82	59.01	56.40	58.83
	CoreRoBERT$_{large}$[45]	57.35	59.43	57.90	60.25
	ATLOP-RoBERTa$_{large}$[18]	61.32	63.18	61.39	63.40
	SSAN$_{large}$[46]	63.76	65.69	63.78	65.92
Transformer+ graph-based	LSR-BERT$_{base}$[20]	52.43	59.00	56.97	59.09
	DISCO-RE$_{base}$[5]	55.91	57.78	55.01	55.70
	GAIN$_{base}$[21]	59.14	61.22	59.00	61.24
	HAG$_{base}$[7]	60.85	63.06	60.78	60.82
	Our BERT$_{base}$	**60.10**	**61.80**	**59.18**	**61.20**
	GAIN$_{large}$[21]	60.87	63.09	60.31	62.76
	Our RoBERTa$_{large}$	**61.61**	**63.41**	**60.95**	**62.96**

Table 3 Comparison of models CDR and GDA for F1- score.

Model		CDR	GDA
		Development/ Test	Development/ Test
Embedding-based	BRAN[47]	– /62.1	– /–
Transformer-based	sciBERT[48]	–/65.1	–/82.5
	SSAN$_{base}$[46]	68.4 /68.7	82.8 /83.7
	ATLOP-sciBERT[18]	–/69.40 (69.29)/(69.44)	–/83.9 (82.49)/(84.45)
Transformer+ grpah-based	EoG[23]	–/63.6	78.7/81.5
	LSR-BERT[20]	– /64.8	– /82.2
	GLRE[49]	–/68.5	–/–
Our SciBERT		69.42 /70.09	82.51/83.63
Our BioBERT		**74.13/ 70.37**	**82.93 /84.52**

Note: The values in brackets are from our reproduction.

and test Ign F1, respectively. For SSAN[46], our model can outperform the indicators on BERT-base. We re-implement ATLOP to further compare with our model. In the same environment, ATLOP outputs a set of values (F1-score: 60.93 and Ign F1: 58.90 on the development set, and F1-score: 60.96 and Ign F1: 58.85 on the test set), in which our results can outperform on all indicators.

4.4.2 Results on CDR and GDA

Table 3 presents our experiment results on the CDR and GDA datasets. They are biomedical texts. Hence, BiomedNLP-PubMedBERT(BioBERT) base[43] is adopted to evaluate our model. From Table 3 our model exceeds all these models on F1-score. We also use the SciBERT base[48] to test our model. Under this situation, our results have a lower value than BioBERT. However, although the test result is worse than those of ATLOP-SciBERT and SSAN$_{base}$, our final results outperform most previous works, showing that BioBERT brings more representation ability to biomedical texts. In summary, our approach is beneficial for this task.

5 Analysis and Discussion

5.1 Ablation study

This work has three critical components: SDT-forest, denoising method, and slide residual attention. To validate the effectiveness of various components, we conduct an ablation study experiment corresponding to different modules:

Model-base is a base model without any components; **SDT-forest** merges the model-base with the novel document-level graph; **Slide residual attention** includes the model-base using a slide residual attention mechanism with the SDT-forest; **Denoising** means the model trims the SDT-forest with the Steiner tree algorithm.

The experimental analysis is performed only on the development set because the DocRED dataset has no ground truth to the test set. Table 4 shows that all modules have a performance increase with each module, indicating that all components contribute to the model performance. The slide residual attention and the denoising method are most important to the model performance and sensitive to the F1-score, leading to an increase of 1.57 (2.6%) and 1.33 (2.3%) in the development F1-score and Ign F1 score when adding these two modules. The denoising method reveals that removing irrelevant words can improve RE performance. Meanwhile, the slide residual attention seems to capture more key information for the document-level RE.

5.2 Case study

Figure 5 presents a visualization of how the SDT-forest transforms into our SG. Figure 5a shows each SDT in one document connected through different dependency relations, syntax, or discourse. Figure 5b is an SG produced by the Steiner tree algorithm, which denoises the irrelevant words to build the long-range semantic dependencies among all entities.

When our model utilizes the Steiner tree algorithm to trim the SDT-forest, most instances in the development set will benefit from this structure. That is, the DGI model will obtain more prediction results. Figure 6 illustrates the incremental results obtained from each instance in the data set, including the true positive and false positive cases. We chose a model[18] without a graphical structure similar to our performance as a baseline. We statistic that 37%(370/1000) of the instances in the DocRED development set will extract more relations than the baseline[18]. This phenomenon shows that the graphical structure can bring more semantics to the model. However, the performance improves insignificantly due to the false positive results. This situation also shows that there should do much work

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

Model	Ign F1	F1-score
Model-base	58.23	60.18
Model-base + SDT-froest	58.77	60.23
Model-base + slide residual-attention	58.92	60.89
Model-base + denoising	60.10	61.80

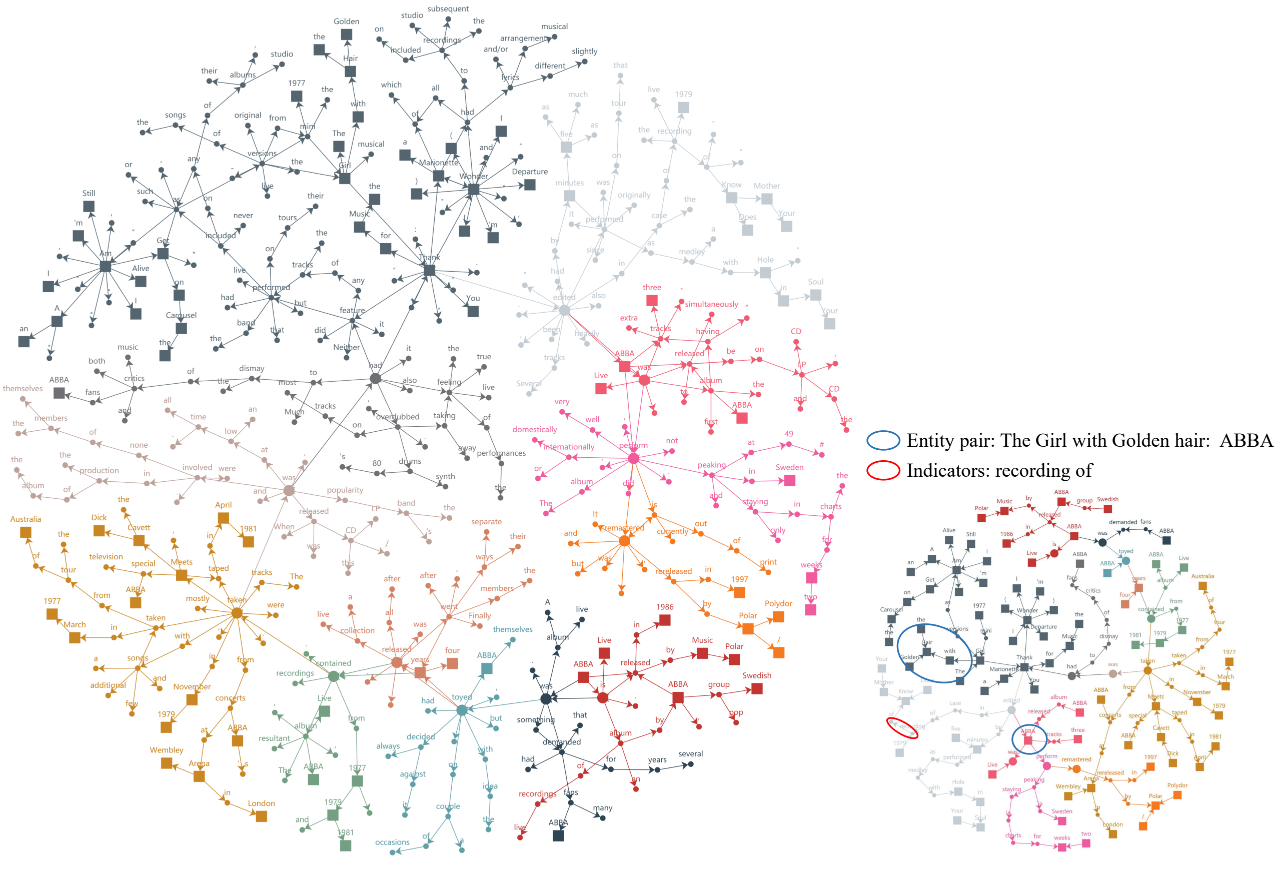


Fig. 5 (a) Example of the SDT-forest combining syntax and discourse dependency relations. (b) Example of the SG from (a) denoising words by the Steiner tree algorithm. The different colour nodes and links indicate different SDTs. The rectangular nodes refer to the entity mention. The others depict words in the document. All nodes connect through syntax, discourse, or "adjacent" relation.

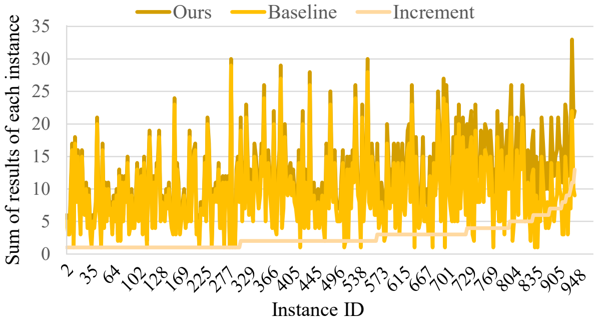


Fig. 6 Comparison of the amount of results of each instance between our model and the baseline[18].

to alleviate this situation.

Table 5 presents examples of the incremental output from our model, and the two rows correspond to two instances. The first row shows four true positive results, which the baseline can not predict. However, this instance has 30 labelled relations in the dataset, but the baseline or our model cannot predict them. Our model and the baseline model predict 21 and 12 relations (including false positive and true positive), respectively. We will analyze them separately from two views, intra-sentence and inter-sentence. We believe that the shortened semantic dependencies between entities (distance between entity pairs) contribute to this situation. Figure 7 depicts an example of the shortened distance between entities in the SG and the original text.

The first case mentions the "Does Your mother Know" recording published in 1979. Note that this is an intra-sentence relation. Both "recording" and "of" are the indicators of this entity pair (Fig. 5b); thus, incorporating them into the graph as evidence words is vital for the prediction. However, the baseline model cannot extract this relation because the sequence-based method rarely highlights the two indicators. From Fig. 5b, our SG links this entity pair via "Does Your Mother Know ⟶recording⟶of⟶1979" (grey colour words), which helps our model predict this relation.

The second case detects the inter-sentence relation

Table 5 Examples of the incremental output from our model.

Entity pair	Relation
Does Your Mother Know: 1979	Publication date
Wembley Arena: London	LATE
The Girl with the Golden Hair: ABBA	Performer
Thank You for the Music: ABBA	Performer
American Airlines Group Inc.: American	Country
Fort Worth: American	Country
Texas: American	Country
Texas: American	LATE
AMR Corporation: American	Country
US Airways Group: American	Country
Federal Aviation Administration: American	A2j
American: Texas	CATE

Note: The above predictions are from our model, which cannot be predicted by the baseline model[18]. These results exclude the false positives or true positives shared by the two models. "LATE", "CATE", and "A2J" mean "Located in the administrative territorial entity", "Contains administrative territorial entity", and "Applies to jurisdiction", respectively.

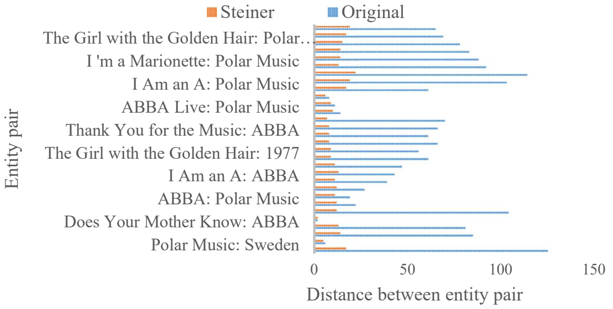


Fig. 7 Distance between the entity pair in the SG and original text.

between mentions "The Girl with the Golden Hair" and "ABBA". However, the entity "ABBA" has multiple mentions, and the nearest mention "ABBA" has a long distance of more than 40 words from the mention "The Girl with the Golden Hair". In Fig. 5b, only four nodes between this entity pair, which shortens the distance between entity pair and builds a long-range semantic dependency across multiple sentences. The second row in Table 5 shows another instance from DocRED, in which our model also predicts more relations than the baseline.

5.3 Analysis of denoising

We analyze our method to show the effectiveness brought by this denoising approach. Figure 8 illustrates the length distribution of the development dataset. The light orange colour means the length of each original text. The dark orange colour depicts the number of SG nodes. Table 6 shows the document length statistics. The original development text's maximum, minimum, and average lengths are 512, 129, and 201, respectively. After our denoising method, the maximum, minimum, and average graph nodes are reduced to 187, 20, and 86, respectively. The maximum length reduces by approximately 63%, the minimum length reduces by 84%, and the average length reduces by 57%.

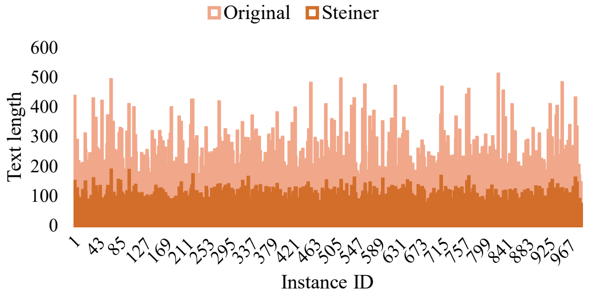


Fig. 8 Text lengths distribution on DocRed.

After denoising, our approach shows better performance for those longer text instances. Figure 9 compares F1-scores between the baseline and our model, and the difference between the two values. We divide the development data into nine intervals of text length and calculate F1-score of the two models in these nine intervals. As shown Fig. 9, the longer the text, the worse the performance, indicating that the number of words significantly affects the performance. However, our

Table 6 Document lengths before and after denoising.

Before denoising			After denoising		
Maximum	Minimum	Average	Maximum	Minimum	Average
512	129	201	187	20	86

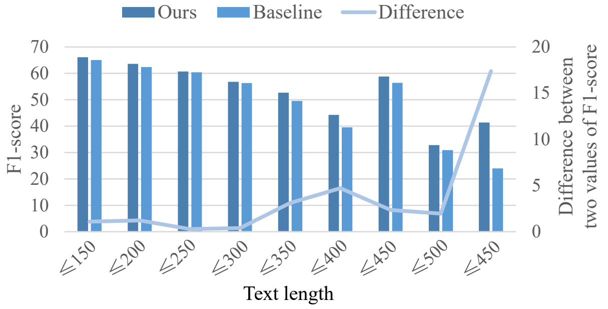
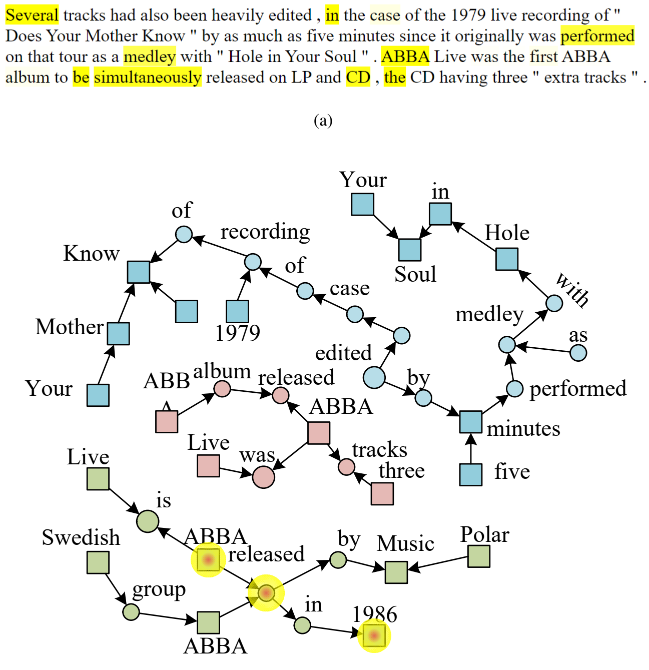


Fig. 9 Comparison of the results from our model and the baseline[18] in terms of the text length on DocRED. The histogram of the text length is divided into nine intervals, [0–150], (150–200], . . . , [500, 550), with the F1-score from the two models. The grey line depicts the difference between two values of F1-score.

model has two characteristics: (1) better performance in each interval than the baseline model, and (2) higher performance gains in longer text. For instance, the different line of the model performance in Fig. 9 shows an upward trend, which means the longer the text, the better the performance brought by our model. In other words, physically reducing the distance between entity pairs may benefit building the long-range semantic dependency for document-level RE. Note that when the text is longer than 400, the unbalanced data distribution causes performance degradation.

5.4 Analysis of attention

This section analyzes the effectiveness brought by the slide residual attention. Figure 10 visualizes the slide residual attention weight on the text and the SG. These attention weights show various features to describe the inter-sentence relational fact (Entity 1: "Does Your Mother Know"; Entity 2: "ABBA", Relation: Performer). Figure 10a shows the attention weights on the text focus on detailing a relation between "Does Your Mother Know" and the live recording "Hole in Your Soul". These yellow words show a few pieces of evidence about the relation between "Does Your Mother Know" and "ABBA". After denoising the graph (see Fig. 10b), the attention on the graph gives a high weight to the word "released", building a long-range semantic dependency along with the graph, which indicates the relation of "ABBA live" (released) between "ABBA". Consequently, the slide residual attention weights gather evidence to infer the ground truth relationship "Performer". His visualization demonstrates that our slide residual attention fusion the words features in context and the node features in the graph that complement each other to predict the relational fact.



Several tracks had also been heavily edited , in the case of the 1979 live recording of " Does Your Mother Know " by as much as five minutes since it originally was performed on that tour as a medley with " Hole in Your Soul " . ABBA Live was the first ABBA album to be simultaneously released on LP and CD , the CD having three " extra tracks " .

(a)

(b)

Fig. 10 (a) Visualization of the attention weight on partial text and (b) attention weight on the SG nodes. The yellow circles are the weights. Subject: "Does Your Mother Know", object: "ABBA", and relation: "Performer".

6 Conclusion

This work proposed a DGI model, which exploits a graphic denoising paradigm to build long-range semantic dependency among all entities, therefore learning an effective representation to reason on text and graph to classify relation categories. First, in the graph construction phrase, we utilized the SDT-forest associating words and sentences through the syntax or discourse relation. Then our denoising approach exploits the Steiner tree to physically reduce linguistically irrelevant words, providing a mention-level denoised graph SG for building the long-range semantic dependency among all entities. Second, after denoising, to fit the training data dispersing in our graph to preserve a better semantic coherence, the slide residual attention gathers information from the context and the graph in two different layers. These attention weights highlight the essential features from two perspectives that complement each other. Third, we conduct extensive experiments on three public datasets. The experiment results show that our model has a competitive result, establishes the long-range semantic dependency among all entities for the document-level RE, and significantly improves the classification performance for those longer text instances.

Acknowledgment

This research work was supported by the National Natural Science Foundation of China (Nos. U19A2059 & 62176046).

References

[1] H. Wang, C. Focke, R. Sylvester, N. Mishra, and W. Wang, Fine-tune Bert for DocRED with two-step process, arXiv preprint arXiv: 1909.11898, 2019.

[2] H. Tang, Y. Cao, Z. Zhang, J. Cao, F. Fang, S. Wang, and P. Yin, HIN: Hierarchical inference network for document-

level relation extraction, in *Proc. of the Advances in Knowledge Discovery and Data Mining*: *24th Pacific-Asia Conf.*, Singapore, 2020, pp. 197–209.

[3] X. Han and L. Wang, A novel document-level relation extraction method based on BERT and entity information, *IEEE Access*, vol. 8, pp. 96912–96919, 2020.

[4] M. Eberts and A. Ulges, An end-to-end model for entity-level relation extraction using multi-instance learning, in *Proc. 16th Conf. European Chapter of the Association for Computational Linguistics: Main Volume*, Virtual event, 2021, pp. 3650–3660.

[5] H. Wang, K. Qin, G. Lu, J. Yin, R. Y. Zakari, and J. W. Owusu, Document-level relation extraction using evidence reasoning on RST-GRAPH, *Knowl.-Based Syst.*, vol. 228, p. 107274, 2021.

[6] C. Yuan, H. Huang, C. Feng, G. Shi, and X. Wei, Document-level relation extraction with Entity-Selection Attention, *Inform. Sci.*, vol. 568, pp. 163–174, 2021.

[7] R. Li, J. Zhong, Z. Xue, Q. Dai, and X. Li, Heterogenous affinity graph inference network for document-level relation extraction, *Knowl.-Based Syst.*, vol. 250, p. 109146, 2022.

[8] R. C. Bunescu and R. J. Mooney, A shortest path dependency kernel for relation extraction, in *Proc. Human Language Technology Conf. and Conf. Empirical Methods in Natural Language Processing*, Vancouver, Canada, 2005, pp. 724–731.

[9] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, Classifying relations via long short term memory networks along shortest dependency paths, in *Proc. 2015 Conf. Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 1785–1794.

[10] P. Gupta, S. Rajaram, H. Schütze, and T. Runkler, Neural relation extraction within and across sentence boundaries, in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 6513–6520, 2019.

[11] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, and M. Sun, DocRED: A large-scale document-level relation extraction dataset, in *Proc. 57th Annu. Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 764–777.

[12] W. C. Mann and S. A. Thompson, Rhetorical structure theory: Toward a functional theory of text organization, *Text-Interdisciplinary Journal for the Study of Discourse*, vol. 8, no. 3, pp. 243– 281, 1988.

[13] Z. Guo, Y. Zhang, Z. Teng, and W. Lu, Densely connected graph convolutional networks for graph-to-sequence learning, *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 297–312, 2019.

[14] K. Xu, Y. Feng, S. Huang, and D. Zhao, Semantic relation classification via convolutional neural networks with simple negative sampling, in *Proc. 2015 Conf. Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 536–540.

[15] Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Y. Lu, and Z. Jin, Improved relation classification by deep recurrent neural networks with data augmentation, in *Proc. COLING 2016, the 26th Int. Conf. Computational Linguistics*: *Technical Papers*, Osaka, Japan, 2016, pp. 1461–1470.

[16] R. Cai, X. Zhang, and H. Wang, Bidirectional recurrent convolutional neural network for relation classification, in *Proc. 54thAnnu. Meeting of the Association for Computational Linguistics* (*Volume 1*: *Long Papers*), Berlin, Germany, 2016, pp. 756–765.

[17] C. Zhang, C. Cui, S. Gao, X. Nie, W. Xu, L. Yang, X. Xi, and Y. Yin, Multi-gram CNN-based self-attention model for relation classification, *IEEE Access*, vol. 7, pp. 5343–5357, 2019.

[18] W. Zhou, K. Huang, T. Ma, and J. Huang, Document-level relation extraction with adaptive thresholding and localized context pooling, *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 16, pp. 14612–14620, 2021.

[19] F. Xue, A. Sun, H. Zhang, J. Ni, and E. S. Chng, An embarrassingly simple model for dialogue relation extraction, in *Proc. of ICASSP 2022–2022 IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*), Singapore, 2022, pp. 6707–6711.

[20] G. Nan, Z. Guo, I. Sekulic, and W. Lu, Reasoning with latent structure refinement for document-level relation extraction, in *Proc. 58th Annu. Meeting of the Association for Computational Linguistics*, Virtual event, 2020, pp. 1546–1557.

[21] S. Zeng, R. Xu, B. Chang, and L. Li, Double graph based reasoning for document-level relation extraction, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing* (*EMNLP*), Virtual event, 2020, pp. 1630–1640.

[22] B. Li, W. Ye, Z. Sheng, R. Xie, X. Xi, and S. Zhang, Graph enhanced dual attention network for document-level relation extraction, in *Proc. 28th Int. Conf. Computational Linguistics*, Barcelona, Spain, 2020, pp. 1551–1560.

[23] F. Christopoulou, M. Miwa, and S. Ananiadou, Connecting the dots: Document-level neural relation extraction with edge-oriented graphs, in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and the 9th International Joint Conf. Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, China, 2019, pp. 4925–4936.

[24] F. Xue, A. Sun, H. Zhang, and E. S. Chng, GDPNet: Refining latent multi-view graph for relation extraction, *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 16, pp. 14194–14202, 2021.

[25] R. C. Bunescu and R. J. Mooney, Subsequence kernels for relation extraction, in *Proc. 18th Int. Conf. Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2005, pp. 171–178.

[26] F. Ren, D. Zhou, Z. Liu, Y. Li, R. Zhao, Y. Liu, and X. Liang, Neural relation classification with text descriptions, in *Proc. 27th Int. Conf. Computational Linguistics*, Santa Fe, NM, USA, 2018, pp. 1167–1177.

[27] H. Wang, K. Qin, G. Lu, G. Luo, and G. Liu, Direction-sensitive relation extraction using Bi-SDP attention model, *Knowl.-Based Syst.*, vol. 198, p. 105928, 2020.

[28] X. Wang, H. Wang, C. Li, T. Huang, and J. Kurths, Improved consensus conditions for multi-agent systems with uncertain topology: The generalized transition rates case, *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1158–1169, 2020.

[29] C. Quirk and H. Poon, Distant supervision for relation extraction beyond the sentence boundary, in *Proc. 15th Conf. European Chapter of the Association for Computational Linguistics* (*Volume 1*: *Long Papers*), Valencia, Spain, 2017, pp. 1171–1182.

[30] N. Peng, H. Poon, C. Quirk, K. Toutanova, and W. T. Yih, Cross-sentence n-ary relation extraction with graph LSTMs, *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 101–115, 2017.

[31] T. Hirao, Y. Yoshida, M. Nishino, N. Yasuda, and M. Nagata, Single-document summarization as a tree knapsack problem, in *Proc. 2013 Conf. Empirical Methods in Natural Language Processing*, Seattle, WA, USA, 2013, pp. 1515–1520.

[32] P. Jansen, M. Surdeanu, and P. Clark, Discourse complements lexical semantics for non-factoid answer reranking, in *Proc. 52nd Annu. Meeting of the Association for Computational Linguistics* (*Volume 1*: *Long Papers*), Baltimore, MD, 2014, pp. 977–986.

[33] Y. Yoshida, J. Suzuki, T. Hirao, and M. Nagata, Dependency-based discourse parser for single-document summarization, in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing* (*EMNLP*), Doha, Qatar, 2014, pp. 1834–1839.

[34] Z. Liu and N. Chen, Exploiting discourse-level segmentation for extractive summarization, in *Proc. 2nd Workshop on New Frontiers in Summarization*, Hong Kong, China, 2019, pp. 116–121.

[35] X. Tan, L. Zhang, F. Kong, and G. Zhou, Towards discourse-aware document-level neural machine translation, in *Proc. 31st Int. Joint Conf. Artificial Intelligence*, Vienna, Austria, 2022, pp. 4383–4389.

[36] W. Zhou and M. Chen, An improved baseline for sentence-level relation extraction, arXiv preprint arXiv: 2102.01373, 2022.

[37] R. Jia, C. Wong, and H. Poon, Document-level n-ary relation extraction with multiscale representation learning, in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1* (*Long and Short Papers*), Minneapolis, MN, 2019, pp. 3693–3704.

[38] Y. Ji and J. Eisenstein, Representation learning for text-level discourse parsing, in *Proc. 52nd Annu. Meeting of the Association for Computational Linguistics* (*Volume 1*: *Long Papers*), Baltimore, MD, USA, 2014, pp. 13–24.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, in *Proc. 31st Int. Conf. Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 6000–6010.

[40] N. Zhang, X. Chen, X. Xie, S. Deng, C. Tan, M. Chen, F. Huang, L. Si, and H. Chen, Document-level relation extraction as semantic segmentation, in *Proc. 30th Int. Joint Conf. Artificial Intelligence*, Montreal, Canada, 2021, pp. 3999–4006.

[41] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C. H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, and Z. Lu, Biocreative V CDR task corpus: A resource for chemical disease relation extraction, *Database*, vol. 2016, p. baw068, 2016.

[42] Y. Wu, R. Luo, H. C. M. Leung, H. F. Ting, and T. W. Lam, RENET: A deep learning approach for extracting gene-disease associations from literature, in *Proc. 23rd Annu. Int. Conf. Research in Computational Molecular Biology*, Washington, DC, USA, 2019, pp. 272–284.

[43] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, Domain-specific language model pretraining for biomedical natural language processing, *ACM Trans. Comput. Healthcare*, vol. 3, no. 1, p. 2, 2022.

[44] Z. Guo, Y. Zhang, and W. Lu, Attention guided graph convolutional networks for relation extraction, in *Proc. 57th Annu. Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 241–251.

[45] D. Ye, Y. Lin, J. Du, Z. Liu, P. Li, M. Sun, and Z. Liu, Coreferential reasoning learning for language representation, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing* (*EMNLP*), Virtual event, 2020, pp. 7170–7186.

[46] B. Xu, Q. Wang, Y. Lyu, Y. Zhu, and Z. Mao, Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction, *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 16, pp. 14149–14157, 2021.

[47] P. Verga, E. Strubell, and A. McCallum, Simultaneously self-attending to all mentions for full-abstract biological relation extraction, in *Proc. 2018 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (*Volume 1*: *Long Papers*), New Orleans, LA, 2018, pp. 872–884.

[48] I. Beltagy, K. Lo, and A. Cohan, SciBERT: A pretrained language model for scientific text, in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, China, 2019, pp. 3615–3620.

[49] D. Wang, W. Hu, E. Cao, and W. Sun, Global-to-local neural networks for document-level relation extraction, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing* (*EMNLP*), Virtual event, 2020, pp. 3711–3721.

Hailin Wang received the MEng and PhD degrees from the University of Electronic Science and Technology of China (UESTC), in 2015 and 2022, respectively. He is currently a lecturer at the School of Computing and Artificial and Intelligence, Southwestern University of Finance and Economics (SWUFE). His current research interests include deep learning and natural language processing.

Ke Qin received the MEng and PhD degrees from UESTC, China in 2006 and 2010, respectively. He is currently a full professor at School of Computer Science and Engineering, UESTC. His research interests include neural networks, machine learning, and natural language, processing.

Guiduo Duan received the BEng degree from Sichuan Normal University, China in 2003, and the MEng and PhD degrees from UESTC, China in 2006 and 2009, respectively. She is now an associate professor at UESTC. She was in the Humboldt University of Berlin as a visiting scholar from 2017 to 2018. Her research interests include deep learning, as well as natural language processing and its applications. She has published more than 20 papers in the international journals and conferences, as well as three books. She has been granted six patents.

Guangchun Luo received the MEng and PhD degrees from UESTC, China in 1999 and 2004, respectively. He is now a professor at UESTC. His research interests include deep learning, as well as decision theory and its applications. He has published more than 60 papers in the international journals and conferences.