HAIN: Hierarchical Aggregation and Inference Network for Document-Level Relation Extraction

Nan Hu, Taolin Zhang, Shuangji Yang, Wei Nong, and Xiaofeng He(✉)

School of Computer Science and Technology, East China Normal University,
Shanghai, China
{51194501046,52184501016,51194501201,51194501160}@stu.ecnu.edu.cn,
hexf@cs.ecnu.edu.cn

Abstract. Document-level relation extraction (RE) aims to extract relations between entities within a document. Unlike sentence-level RE, it requires integrating evidences across multiple sentences. However, current models still lack the ability to effectively obtain relevant evidences for relation inference from multi-granularity information in the document. In this paper, we propose **H**ierarchical **A**ggregation and **I**nference **N**etwork (HAIN), performing the model to effectively predict relations by using global and local information from the document. Specifically, HAIN first constructs a *meta dependency graph* (mDG) to capture rich long distance global dependency information across the document. It also constructs a *mention interaction graph* (MG) to model complex local interactions among different mentions. Finally, it creates an *entity inference graph* (EG), based on which we design a novel hybrid attention mechanism to integrate relevant global and local information for entities. Experimental results demonstrate that our model achieves superior performance on a large-scale document-level dataset (DocRED). Extensive analyses also show that the model is particularly effective in extracting relations between entities across multiple sentences and mentions.

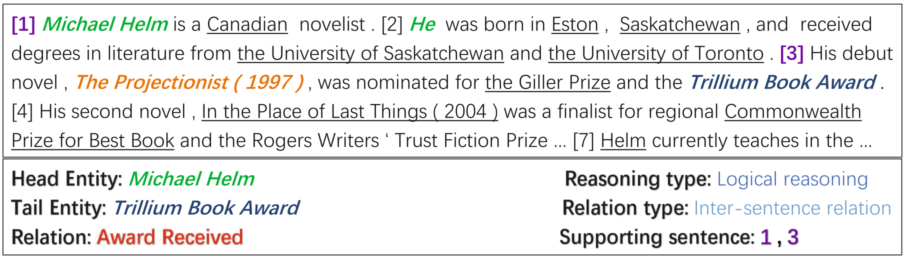
Keywords: Document-level relation extraction · Graph neural network

1 Introduction

Relation extraction (RE) aims to identify semantic relations between entities from plain text. With the growing demand for structured knowledge, RE has attracted much attention in natural language processing. Prior works have made great progress in extracting relations within a sentence (sentence-level RE). However, in real world scenarios, a large number of relation instances appear across sentences. Compared with sentence, a document often contains many entities, and some entities have multiple mentions under the same phrase of alias. Hence, document-level RE is a more complex relation extraction problem.

L. Wang et al. (Eds.): NLPCC 2021, LNAI 13028, pp. 325–337, 2021.
https://doi.org/10.1007/978-3-030-88480-2_26

Figure 1 shows an example of document-level RE. Early studies [10,12] defined document-level RE to short text spans (e.g., document only contains two sentences). Some other studies were limited to specific domain (e.g., biomedicine). It's obviously that they are incapable of dealing with the example in Fig. 1. Recent works [1,7,9] used graph-based neural approaches, since graph has proven useful in encoding long distance, cross-sentential information. They mainly put different types of nodes in a same graph and then applied vanilla GCNs [6] to jointly update nodes. However, current models do not in-depth explore a reasonable graph aggregation and inference structure which is critical to model's understanding of the entire document.



[1] ***Michael Helm*** is a Canadian novelist . [2] ***He*** was born in Eston , Saskatchewan , and received degrees in literature from the University of Saskatchewan and the University of Toronto . **[3]** His debut novel , ***The Projectionist (1997)*** , was nominated for the Giller Prize and the ***Trillium Book Award*** . [4] His second novel , In the Place of Last Things (2004) was a finalist for regional Commonwealth Prize for Best Book and the Rogers Writers ' Trust Fiction Prize ... [7] Helm currently teaches in the ...

Head Entity: ***Michael Helm***
Tail Entity: ***Trillium Book Award***
Relation: **Award Received**

Reasoning type: Logical reasoning
Relation type: Inter-sentence relation
Supporting sentence: **1 , 3**

Fig. 1. An example from the DocRED [20] dataset. Entities and mentions involved in the relation instance (*Michael Helm*, *Award Received*, *Trillium Book Award*) are colored. Other irrelevant mentions are underlined for clarity (best viewed in color).

From our point of view, as Fig. 1 shows, in order to extract the relation between *Michael Helm* and *Trillium Book Award*. Firstly, we should identify sentence 1 and 3 are supporting sentences that contain the global context information about *Michael Helm* and *Trillium Book Award*. Then, identify *Michael Helm* is a novelist from sentence 1, *The Projectionist(1997)* is a novel written by *Michael Helm* and nominated for *Trillium Book Award* from sentence 3. Finally, we can infer that *Michael Helm* received *Trillium Book Award*. Obviously, it's a step by step inference behavior, multi-granularity information is aggregated from coarse to fine (document → mention → entity). But the supporting sentences are scattered in the document, relevant mentions usually don't appear in the same sentence, and entities need long distance dependency information.

In this paper, we propose a novel graph-based network for document-level RE. Our primary motivation is to design a hierarchical aggregation and inference structure that can do document-level RE as the above intuitive example. Towards this goal, we address three challenges: (1) *how to capture long distance dependency information of a document?* Syntactic dependency tree conveys rich structural information that is proven useful for many sentence-level RE models [4,23]. We extend it to document-level, and build a meta dependency graph (mDG) that can utilize structural knowledge to capture long distance global dependency information of a document. (2) *how to model complex local information of mentions?* We construct a mention interaction graph (MG) to capture local information by mention interactions. Concretely, we merge the initial representations of mentions from mDG, build MG by self-attention mechanism [17] and then apply GCN [6] to encode MG. (3) *how to learn entity representations effectively?* We build an entity inference graph (EG) and design a novel hybrid attention mechanism to encode global and local information from mDG and MG into entities.

Our main contributions can be summarized as follows:

1. We propose a Hierarchical Aggregation and Inference Network (**HAIN**), which features a hierarchical graph design, to better cope with document-level RE task.
2. We introduce three different graphs to meet the needs of different granularity information. A novel hybrid attention mechanism is proposed to effectively aggregate global and local information for entities.
3. HAIN achieves new state-of-the-art performance on DocRED dataset. Our detailed analysis further shows its superior advantage in extracting relations between entities of long distance.

2 Methodology

2.1 Model Overview

Given a document D $= [x_1, x_2, ..., x_n]$, where $i \in [1, n]$ and x_i is the i-th word in document. Sentences, entities and their corresponding textual mentions are annotated in the document. The set of relation types is pre-defined. Our goal is to identify the relations of all entity pairs in the document. Obviously, it is a multi-label classification problem.

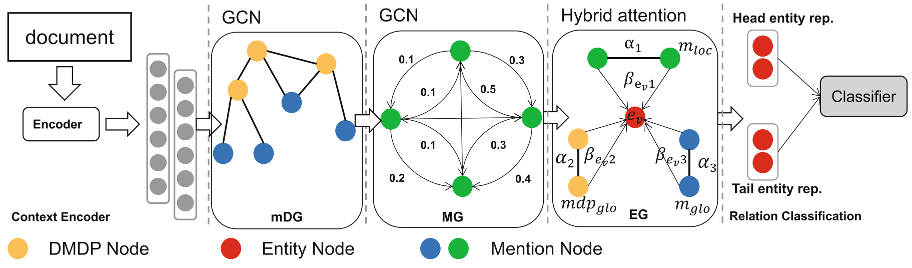


Fig. 2. Architecture of HAIN. Some nodes are omitted for simplicity. MG is a fully connected graph with learned edge weight from 0.0 to 1.0. In EG, α_i, $\beta_{e_v i}$ are type and node attention scores of entity e_v calculated by hybrid attention mechanism. m_{loc} is mention nodes representations learned from MG, m_{glo}, mdp_{glo} are mention and DMDP nodes representations learned from mDG.

Figure 2 depicts the architecture of our HAIN. (1) First, it uses LSTM [14] or BERT [2] as encoder to receive an entire document with annotations as input and output the contextual representation of each word. (2) Next, it constructs a *meta dependency graph* (mDG) by using the dependencies of the syntactic dependency tree. It also creates a *mention interaction graph* (MG) by self-attention mechanism [17]. mDG and MG graphs are encoded by using stacked GCN [6] to respectively capture global and local information of the document. (3) Then, a novel hybrid attention mechanism is designed to integrate relevant global and local relation inference information into entities in entity inference graph (EG). (4) Finally, it uses entities representations learned from EG to predict relations.

2.2 Context Encoder

To obtain the contextual representation of each word, we feed a document D into a contextual encoder. The context encoder can be a bidirectional LSTM [14] or BERT [2]. Here we use the BiLSTM as an example:

$$\overleftarrow{h_{w_j}} = \mathbf{LSTM}(\overleftarrow{h_{w_{j+1}}}, \gamma_j) \tag{1}$$

$$\overrightarrow{h_{w_j}} = \mathbf{LSTM}(\overrightarrow{h_{w_{j-1}}}, \gamma_j) \tag{2}$$

where $\overleftarrow{h_{w_j}}$ and $\overrightarrow{h_{w_j}}$ represent the hidden representations of the j-th word in the document of two directions, γ_j indicates the word embedding of the j-th word. Finally, the contextual representation of each word in the document is represented as $h_{w_j} = [\overleftarrow{h_{w_j}}; \overrightarrow{h_{w_j}}]$.

2.3 Meta Dependency Graph

Based on the contextual representation of each word, we extract document meta dependency path nodes (DMDP) and mention nodes to construct *meta dependency graph*. The initial representation of a mention node $\mathbf{m}_i$ is calculated by averaging the representations of contained words (e.g., $h_{m_i} = [avg_{w_j \in m_i}(h_{w_j})]$). Early approaches [4,13] used all nodes in the syntactic dependency tree of a sentence. Nan et al., [9] just extracted nodes on the shortest dependency path (SMDP) between mentions in the sentence, as it is able to make full use of relevant information while ignoring irrelevant information. We extend it to DMDP by connecting root nodes of each sentence dependency tree in a document.

As Fig. 3 shows, given four mentions m_1, m_2, m_3, m_4 in two sentences s_1, s_2 of document D, and $m_1, m_2 \in s_1$, $m_3, m_4 \in s_2$. SMDP just extracts MDP_{m_1,m_2} and MDP_{m_3,m_4} as nodes. But our DMDP extracts MDP_{m_i,m_j}, $i, j \in 1, 2, 3, 4$ and $i \neq j$ as nodes, which will contain more inter-sentential information.

We define an adjacency matrix $\mathbf{A_D}$ to represent the *meta dependency graph*, where $\mathbf{A_D}_{i,j} = 1$ when there is an edge connects node i and node j in dependency tree. Then we employ a L-layer stacked GCN [6] to convolute the *meta dependency graph*. Given node $\mathbf{u}$ at the l-th layer, the graph convolutional operation can be defined as:

$$h_{\mathbf{u}}^{(l+1)} = RELU\left(\sum_{j=1}^{n} \mathbf{A_D}_{i,j} \mathbf{W}^{(l)} h_{\mathbf{u}_j}^{(l)} + b^{(l)}\right) \tag{3}$$

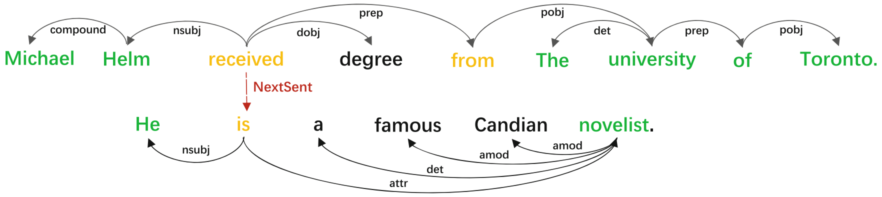


Fig. 3. An example of document meta dependency path nodes (DMDP). Mention and DMDP nodes are respectively colored in green and yellow. (Color figure online)

where $\mathbf{W}^{(l)} \in \mathbb{R}^{d_n \times d_n}$ and $b^{(l)} \in \mathbb{R}^{d_n}$ are trainable parameters, d_n is the dimension of node representations.

After the graph information propagation in *meta dependency graph*, we can obtain new representations of mention and DMDP nodes, we respectively denote them by $\mathbf{m}_{glo}$ and $\mathbf{mdp}_{glo}$ which encode the semantic information of the whole document.

2.4 Mention Interaction Graph

Past works [4,9,18] showed that local information is also important for relation classification, which can be captured by mention interactions. But the local context of different mentions is complex, it is hard to create a graph by explicit rules (e.g., co-references, syntactic trees or heuristics). Hence, we employ soft-attention mechanism [17] to construct an implicit graph. The key idea is to use attention for inducing interactions between mention nodes, especially for those connected by indirect, multi-hop paths.

We first compute an adjacency matrix $\mathbf{A_M}$ for *mention interaction graph* by using self attention mechanism [17]. Then similar to previous steps in mDG, we apply graph convolutional operation to aggregate mention interactions.

$$\mathbf{A_M} = softmax(\frac{QW_t^Q \times (KW_t^K)^T}{\sqrt{d_n}}) \tag{4}$$

$$h_m^{(l+1)} = RELU\left(\sum_{j=1}^{n} \mathbf{A_M}_{ij} \mathbf{W}^{(l)} h_{m_j}^{(l)} + b^{(l)}\right) \tag{5}$$

where $W^Q \in \mathbb{R}^{d_n \times d_n}, W^K \in \mathbb{R}^{d_n \times d_n}$ are trainable projection matrices. Q and K are both equal to $\mathbf{m}_{glo}$ which is from mDG. $\mathbf{W}^{(l)} \in \mathbb{R}^{d_n \times d_n}$ and $b^{(l)} \in \mathbb{R}^{d_n}$ are trainable parameters. After the operation of mutual reasoning between mentions, we get the mention representations $\mathbf{m}_{loc}$, which contain local information of mentions.

2.5 Entity Inference Graph

The goal of *entity inference graph* is to integrate long distance global information from mDG, and local interaction information from MG into entities. Therefore, we generate a fully connect weighted graph with $\mathbf{mdp}_{glo}$, $\mathbf{m}_{glo}$, $\mathbf{m}_{loc}$ and $\mathbf{e}$ nodes. The initial representation of an entity node $\mathbf{e}_i$ is calculated by averaging of its mention representations (e.g., $h_{e_i} = [avg_{m_j \in e_i}(h_{m_j})]$).

Given a specific entity $\mathbf{e}_i$, different types of neighboring nodes may have different impacts on it. For example, the $\mathbf{mdp}_{glo}$ may contain more inter-sentential global information than $\mathbf{m}_{loc}$. But when $\mathbf{e}_i$ needs fine-grained information, $\mathbf{m}_{loc}$ is more useful. Additionally, different neighboring entities could also have different importance. To capture both the different importance at neighboring node level and neighboring type level for entities, we design a novel hybrid attention mechanism which can learn the graph connection weights in end to end fashion.

Neighboring Type Attention. For an entity node $\mathbf{e}_v$, the neighboring type attention learns the weights of different types of neighboring nodes. Specifically, we first represent the embedding of the type τ as $h_\tau = \sum_{v' \in \mathcal{N}_{e_v}} h_{v'}$, which is the sum of the neighboring node features $h_{v'}$, where the nodes $v' \in \mathcal{N}_{e_v}$ and are with the type τ. Then, we calculate the type attention scores based on the current node embedding h_{e_v} and the type embedding h_τ:

$$a_\tau = LeakyRELU(\mu_\tau^T \cdot [h_{e_v} || h_\tau]) \tag{6}$$

where μ_τ is the trainable attention vector for the type τ.

Then we obtain the type attention weights by normalizing the attention scores across all the types with the softmax function:

$$\alpha_\tau = \frac{exp(a_\tau)}{\sum_{\tau' \in \mathcal{T}} exp(a_{\tau'})} \tag{7}$$

Neighboring Node Attention. We design the neighboring node attention to capture the importance of different neighboring nodes and reduce the weights of noisy nodes. Formally, for entity node e_v and its neighboring node $v' \in \mathcal{N}_{e_v}$ with the type τ', we compute the node attention scores based on the node embeddings h_{e_v} and $h_{v'}$ with the type attention weight $\alpha_{\tau'}$ for the node v':

$$\beta_{e_v v'} = \sigma(v^T \cdot \alpha_{\tau'} [h_{e_v} || h_{v'}]) \tag{8}$$

where v is the trainable attention vector. Then we normalize the node attention scores similar to above:

$$\beta'_{e_v v'} = \frac{exp(\beta_{e_v v'})}{\sum_{u \in \mathcal{N}_{e_v}} exp(\beta_{e_v u})} \tag{9}$$

After the computation of type attention and node attention, the representations of all neighboring nodes h_u in $\mathcal{N}_{e_v}$ are aggregated to $\bar{h}'_{e_v}$:

$$\bar{h}_{e_v} = RELU(\sum_{u \in \mathcal{N}_{e_v}} \beta'_{e_v v'} (h_u \mathbf{W}_v + b_v)) \tag{10}$$

$$\bar{h}'_{e_v} = \mathcal{LN}\left(\bar{h}_{e_v} + \left(\sigma\left(\bar{h}_{e_v} \mathbf{W}_{l1} + b_{l1}\right) \mathbf{W}_{l2}\right)\right) \tag{11}$$

where $\mathbf{W}_v \in \mathbb{R}^{d_n \times d_n}, \mathbf{W}_{l1} \in \mathbb{R}^{d_n \times 4d_n}, \mathbf{W}_{l2} \in \mathbb{R}^{4d_n \times d_n}$. $b_v \in \mathbb{R}^{d_n}$ and $b_{l1} \in \mathbb{R}^{4d_n}$ are the bias vectors. $\mathcal{LN}$ is the LayerNorm function and $\sigma(\cdot)$ is activation function GELU. $\bar{h}'_{e_v}$ is the v-th entity representation from EG. We get the final representation $\mathbf{e}$, which contains a vast amount of relation inference information.

2.6 Relation Classification

To classify the relations for an entity pair $(\mathbf{e}^{head}, \mathbf{e}^{tail})$, we first concatenate entity representations and relative distance representations as follows:

$$\hat{\mathbf{e}}^{head} = [\mathbf{e}^{head}; \mathbf{Dist}(\delta_{ht})] \tag{12}$$

$$\hat{\mathbf{e}}^{tail} = [\mathbf{e}^{tail}; \mathbf{Dist}(\delta_{th})] \tag{13}$$

where δ_{ht} means the relative distance of the head entity to tail entity, δ_{th} is similarly defined. **Dist** is a trainable relative distance embedding matrix. Then, we use a bilinear function to compute the probability for each relation type:

$$P(r|\mathbf{e}^{head}, \mathbf{e}^{tail}) = sigmoid(\mathbf{W}_{r_2}\sigma(\hat{\mathbf{e}}^{head}\mathbf{W}_{r_1}\hat{\mathbf{e}}^{tail} + b_{r_1}) + b_{r_2}) \tag{14}$$

where $\mathbf{W}_{r_1}, \mathbf{W}_{r_2} \in \mathbb{R}^{d_n \times d_n \times d_r}, b_{r_1}, b_{r_2} \in \mathbb{R}^{d_r}$ are relation type dependent trainable parameters, d_r is the number of relation types. We use binary cross entropy as the classification loss to train HAIN:

$$loss = -\sum_{r=1}^{d_r} y_r log\, P(r|\mathbf{e}^{head}, \mathbf{e}^{tail})) + (1 - y_r) log(1 - P(r|\mathbf{e}^{head}, \mathbf{e}^{tail})) \tag{15}$$

where $y_r \in \{0, 1\}$ is the true value on relation r.

3 Experiments

3.1 Dataset

We evaluate HAIN on DocRED [20] builted from Wikipedia and Wikidata, which is the largest document-level RE dataset. Both human-annotated and distantly-supervised data are offered. We only use the human-annotated data.

3.2 Baseline Models

We compare our HAIN with the following models.

- **Sequence-based Models.** Yao et al. [20] proposed several baseline models which used CNN/LSTM as encoder and predicted relations between entities by a bilinear function. Context-Aware [15] incorporated context relation information by attention, and Yao et al. [20] adapted it for document-level RE. HIN [16] aggregated the inference information of different granularity to predict relations.

- **Graph-based Models.** LSR [9] induced a latent document graph by maximum tree theory and used GCN for multi-hop reasoning. Nan et al. [9] also adopted GCNN [13] and AGGCN [23] for DocRED, while these are state-of-the-art sentence-level RE models. GEDA [7] characterized the complex interaction between sentences via a dual attention network. GAIN [22] proposed a novel path reasoning mechanism to infer relations between entities.
- **PLM-based Models.** BERT-RE [19] simply used BERT [2] as encoder to get a contextual entity representations. CorefBERT [21] designed a mention reference prediction task to enhance the coreferential reasoning ability of the pre-trained language model explicitly.

Table 1. Main results of different models on DocRED. Results with † are implemented and published by Nan et al. [9]. Other results are reported in their original papers.

Model	Dev		Test	
	IgnF1	F1	IgnF1	F1
CNN [20]	41.58	43.45	40.33	42.26
LSTM [20]	48.44	50.68	47.71	50.07
Context-Aware [20]	48.94	51.09	48.40	50.70
HIN-GloVe [16]	51.06	52.95	51.15	53.30
GCNN† [13]	46.22	51.52	49.57	51.62
AGGCN† [3]	46.29	52.47	48.89	51.45
GEDA [7]	51.03	53.60	51.22	52.97
LSR-GloVe [9]	48.82	55.17	52.15	54.18
GAIN-GloVe [22]	53.05	55.29	52.66	55.08
HAIN-GloVe	**54.98**	**56.03**	**54.73**	**55.76**
BERT-RE$_{base}$ [19]	–	54.16	–	53.20
GEDA-BERT$_{base}$ [7]	54.52	56.16	53.71	55.74
HIN-BERT$_{base}$ [16]	54.29	56.31	53.70	55.60
CorefBERT$_{base}$ [21]	55.32	57.51	54.54	56.96
LSR-BERT$_{base}$ [9]	52.43	59.00	56.97	59.05
GAIN-BERT$_{base}$ [22]	59.14	61.22	59.00	61.24
HAIN-BERT$_{base}$	**59.77**	**62.31**	**59.43**	**61.41**
CorefBERT$_{large}$ [21]	56.73	58.88	56.48	58.70
GAIN-BERT$_{large}$ [22]	60.87	63.09	60.31	62.76
HAIN-BERT$_{large}$	**61.27**	**63.91**	**61.23**	**63.01**

3.3 Experimental Setup

Following Yao et al. [20], we use the GloVe [11] embedding with BiLSTM, and BERT [2] as the context encoder. We use spaCy[1] to get syntactic dependency

[1] https://spacy.io/.

parse tree for each sentence. Then we use NetWorkX[2] to represent the dependency parse tree. In our HAIN implementation, we use 3 layers of GCN and set the dropout rate to 0.4, learning rate to 0.001. We train HAIN using Adam [5] as optimizer. All hyper-parameters are tuned on the development set.

We use F1 as the evaluation metric. Due to some relation instances are present in both training and dev/test sets, to avoid introducing evaluation bias, we also report Ign F1 which denotes F1 scores excluding relation instances shared by the training and dev/test sets.

3.4 Main Results

Table 1 lists the results of different models in DocRED [20] dev and test set. We can find that:

(1) The graph-based models [3,9] obtain comparable results, and the best graph-based model LSR [9] outperforms the best sequence-based model HIN [16]. We owe it to the graph structure can better encode long distance, cross-sentential information. (2) BERT [2] can further boost the performance of our model, which indicates the importance of prior knowledge. For example, HAIN-BERT$_{base}$ outperforms HAIN-GloVe 6.28/5.65 in F1 scores. (3) HAIN-BERT$_{large}$ has achieved the best results compared with all the models. We attribute it to the hierarchical graph structure and hybrid attention mechanism, the former can model global and local information from the document, the latter can effectively synthesize them.

Table 2. Intra- and inter-sentence experimental results. (Models with ♠ are reported in Nan et al., [9]. Model with † is re-trained based on their open implementation.)

Model	Intra-F1	Inter-F1
LSTM ♠ [20]	56.57	41.47
LSR-GloVe ♠ [9]	60.83	48.35
GAIN-GloVe [22]	61.67	48.77
HAIN-GloVe	**62.72**	**49.87**
BERT-RE$_{base}$ ♠ [19]	61.61	47.15
GLRE† [18]	63.63	51.56
LSR-BERT$_{base}$ ♠ [9]	65.26	52.05
GAIN-BERT$_{base}$ [22]	67.10	53.90
HAIN-BERT$_{base}$	**68.34**	**54.70**

[2] https://networkx.org/.

3.5 Detail Analysis

Intra- and Inter-sentence Performance. An entity pair requires inter-sentence reasoning if the two entities from the same document have no mentions in the same sentence. We report the Intra-F1 and Inter-F1 scores in Table 2, which only consider intra- or inter-sentence relations respectively.

Under the same setting, our HAIN outperforms all the other models in both intra- and inter- sentence setting. In particular, the differences in Inter-F1 scores between HAIN and other models tend to be larger than the differences in the Intra-F1 scores. For example HAIN-BERT$_{base}$ improves 2.65 Inter-F1 scores compared with LSR-BERT$_{base}$. The results suggest that the hierarchical aggregation and inference structure of our model is capable of integrating the information across long distance, multiple sentences of a document.

Ablation Study. To further analyze HAIN, we conduct some ablation studies to verify the effectiveness of different modules and mechanisms of HAIN. Results are shown in Table 3. We can observe that: (1) When we remove DMDP nodes, and use SMDP nodes as Nan et al., [9], Inter-F1 drops by 1.26 scores. It means that DMDP nodes can capture richer inter-sentential information than traditional SMDP nodes. (2) F1 and Inter-F1 drops when we remove *meta dependency graph*, it shows that mDG can capture long distance dependency information. (3) Taking away *mention interaction graph*, Intra-F1 sharply drops by 4.59 scores. This drop shows that MG plays a vital role in capturing local information. (4) We remove the Hybrid attention mechanism. To be specific, we directly use the original GCN [6] to convolute the *entity inference graph*, ignoring the different importance of multi-granularity information. The Hybrid attention mechanism's removal results in poor performance across all metrics. It suggests that our hybrid attention mechanism helps aggregate global and local information, therefore, improve the overall performance of document-level RE.

Table 3. Ablation Study of HAIN-BERT$_{base}$ on DocRED dev set.

	F1	Ign F1	Intra-F1	Inter-F1
Full model	**62.31**	**59.77**	**68.34**	**54.70**
– DMDP Node	59.33	58.97	67.46	53.44
– Meta dependency graph	58.40	59.66	67.01	53.87
– Mention interaction graph	58.23	59.07	63.75	53.90
– Hybrid attention mechanism	57.89	56.23	60.77	51.10

Case Study. We list a few examples from DocRED dev set in Table 4, and use HAIN-GloVe in comparison with GAIN-GloVe [22] which is one of the most powerful graph-based model recently. We can observe that: (1) From example 1, we can find that long distance dependency information is necessary. The head entity *William Earl Barber* and tail entity *Marines* cross five sentences, which need the model to be robust enough to tackle long distance cross sentence information. HAIN can capture long distance dependency information by meta dependency graph (mDG) to correctly identify the relation *military branch*. (2) From example 2, we can observe that logical reasoning is vital. We know *Dany Morin* is a *Canadian* in sentence 1, *Dany Morin* is a member of *New Democratic Party* in sentence 2. Extracting the relation between *Canadian* and *New Democratic Party* needs the bridge entity *Dany Morin*. HAIN handled this problem by reasoning in the entity inference graph (EG), which can fuse global and local important information to capture the logical relations. (3) Commonsense knowledge is required in example 3. Models must know that *M* is the code name of a person ahead of time, then identify the relation of *Miss Moneypenny* and *Bond* is *present in work*. Both HAIN and GAIN can not solve this issue, due to lack of the commonsense knowledge. We leave it as our future work.

Table 4. Case study on the DocRED. **Head entities** and **Tail entities** are colored accordingly. Other relevant entities are colored in blue.

[1] **William Earl Barber** (November 30 , 1919 April 19 , 2002) was a United States Marine Corps colonel. [2] **He** fought on Iwo Jima during World War II and was awarded the Medal of Honor for his actions in the Battle of Chosin Reservoir during the Korean War ... [4] Despite the extreme cold weather conditions and a bullet wound to the leg, Barber refused evacuation and an order for his company to ... [5] Barber, aware that leaving would cause 8,000 **Marines** of his division to be trapped in North Korea, held on to the position with his men ...		
Relation Label: military branch	**HAIN:** military branch	**GAIN:** N/A
[1] Dany Morin (born December 19, 1985) is a **Canadian** businessman and former politician. [2] He represented the electoral district of Chicoutimi: Le Fjord as a member of the **New Democratic Party** ... [3] He served as the NDP associate critic for lesbian, gay, bisexual, transgender, and transsexual issues, alongside lead critic Randall Garrison ...		
Relation Label: country	**HAIN:** country	**GAIN:** N/A
[1] **Miss Moneypenny**, later assigned the first names of Eve or Jane , is a fictional character in the James Bond novels and films. [2] She is secretary to M, who is Bond 's superior officer and head of the British Secret Intelligence Service (MI6). [3] Although she has a small part in most of the films, it is always highlighted by the underscored romantic tension between her and **Bond** ...		
Relation Label: present in work	**HAIN:** N/A	**GAIN:** N/A

4 Related Work

In practice, many real world relation instances can only be extracted across sentences. For example, Yao et al., [20] made an analysis on Wikipedia corpus, at least 40.7% of relations can only be extracted on the document level. Therefore, natural language processing community has gradually pay much attention to document-level RE. To accelerate the research on document-level RE, Yao et al. [20] introduced DocRED, constructed from Wikipedia and Wikidata. At present, DocRED is the largest document-level RE dataset. Quirk et al., [12] incorporated both standard dependencies and discourse relations in RE. Peng et al., [10] explored different LSTM approaches with various dependencies, such as syntactic and sequential. But they both captured document specific features, ignored relational inference in document. Recently, many graph-based models are designed to handle this problem. Sahu et al., [13] utilized syntactic parsing and coreference resolution to build a document-level graph for graph inference. Christopoulou et al., [1] constructed a document graph with heterogeneous types of nodes and edges, and proposed edge-oriented model for global relation inference. Li et al., [7] proposed a dual attention network to characterize the interactions in document. Nan et al., [9] treated the graph structure as a latent variable and constructed it by utilizing structured attention [8]. Zeng et al. [22] proposed a novel path reasoning mechanism to enhance the reasoning abilities for RE. Different from the previous works, we construct a hierarchical graph which can utilize the structural information from syntactic trees to capture long-distance dependency. Moreover, we propose a novel hybrid attention mechanism to effectively aggregate global and local information to reason logical relations between entities.

5 Conclusion

In this paper, we proposed a hierarchical aggregation and inference network (HAIN) for document-level RE. It respectively establishes three different information granularity graphs which can effectively integrate relevant relation inference evidences from coarse to fine. Experiments show that our HAIN achieves state-of-the-art performance on the widely used dataset DocRED. In the future, we plan to utilize extra commonsense knowledge to help train more efficient models for solving the commonsense relation inference problem.

References

1. Christopoulou, F., Miwa, M., Ananiadou, S.: Connecting the dots: document-level neural relation extraction with edge-oriented graphs. In: EMNLP (2019)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
3. Guo, Z., Zhang, Y., Lu, W.: Attention guided graph convolutional networks for relation extraction. In: ACL (2019)
4. Gupta, P., Rajaram, S., Schütze, H., Runkler, T.: Neural relation extraction within and across sentence boundaries. In: AAAI (2019)
5. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
6. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
7. Li, B., Ye, W., Sheng, Z., Xie, R., Xi, X., Zhang, S.: Graph enhanced dual attention network for document-level relation extraction. In: Coling (2020)
8. Liu, Y., Lapata, M.: Learning structured text representations. In: TACL (2018)
9. Nan, G., Guo, Z., Sekulić, I., Lu, W.: Reasoning with latent structure refinement for document-level relation extraction. In: ACL (2020)

10. Peng, N., Poon, H., Quirk, C., Toutanova, K., Yih, W.T.: Cross-sentence N-ary relation extraction with graph LSTMs. In: TACL (2017)
11. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
12. Quirk, C., Poon, H.: Distant supervision for relation extraction beyond the sentence boundary. In: EACL (2016)
13. Sahu, S.K., Christopoulou, F., Miwa, M., Ananiadou, S.: Inter-sentence relation extraction with document-level graph convolutional neural network. In: ACL (2019)
14. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Sig. Process. **45**(11), 2673–2681 (1997)
15. Sorokin, D., Gurevych, I.: Context-aware representations for knowledge base relation extraction. In: EMNLP (2017)
16. Tang, H., et al.: HIN: hierarchical inference network for document-level relation extraction. In: PAKDD (2020)
17. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
18. Wang, D., Hu, W., Cao, E., Sun, W.: Global-to-local neural networks for document-level relation extraction. In: EMNLP (2020)
19. Wang, H., Focke, C., Sylvester, R., Mishra, N., Wang, W.: Fine-tune BERT for DocRED with two-step process. arXiv preprint arXiv:1909.11898 (2019)
20. Yao, Y., et al.: DocRED: a large-scale document-level relation extraction dataset. In: ACL (2019)
21. Ye, D., Lin, Y., Du, J., Liu, Z., Sun, M., Liu, Z.: Coreferential reasoning learning for language representation. In: EMNLP (2020)
22. Zeng, S., Xu, R., Chang, B., Li, L.: Double graph based reasoning for document-level relation extraction. In: EMNLP (2020)
23. Zhang, Y., Guo, Z., Lu, W.: Attention guided graph convolutional networks for relation extraction. In: ACL (2019)