Double Graph Based Reasoning for Document-level Relation Extraction

Shuang Zeng[1,2]*, **Runxin Xu**[1]*, **Baobao Chang**[1,2]† and **Lei Li**[3]
[1]Key Laboratory of Computational Linguistics, Peking University, MOE, China
[2]School of Software and Microelectronics, Peking University, China
[3]ByteDance AI Lab, China
{zengs,chbb}@pku.edu.cn runxinxu@gmail.com lileilab@bytedance.com

Abstract

Document-level relation extraction aims to extract relations among entities within a document. Different from sentence-level relation extraction, it requires reasoning over multiple sentences across a document. In this paper, we propose **G**raph **A**ggregation-and-**I**nference **N**etwork (GAIN) featuring double graphs. GAIN first constructs a *heterogeneous mention-level graph* (hMG) to model complex interaction among different mentions across the document. It also constructs an *entity-level graph* (EG), based on which we propose a novel path reasoning mechanism to infer relations between entities. Experiments on the public dataset, DocRED, show GAIN achieves a significant performance improvement (2.85 on F1) over the previous state-of-the-art. Our code is available at https://github.com/DreamInvoker/GAIN.

Elias Brown
[1] *Elias Brown* (*May 9, 1793*– *July 7, 1857*) was a ***U.S.*** Representative from ***Maryland***. [2] Born near ***Baltimore***, ***Maryland***, *Brown* attended the common schools. ... [7] He died near ***Baltimore***, ***Maryland***, and is interred in a private cemetery near ***Eldersburg***, ***Maryland***.

Subject: ***Maryland***
Object: ***U.S.***
relation: **country**

Subject: ***Baltimore***; ***Eldersburg***
Object: ***Maryland***
relation: **located in the administrative territorial entity**

Subject: ***Baltimore***; ***Eldersburg***
Object: ***U.S.***
relation: **country**

Figure 1: An example document and its desired relations from DocRED (Yao et al., 2019). Entity mentions and relations involved in these relation instances are colored. Other mentions are underlined for clarity.

1 Introduction

The task of identifying semantic relations between entities from text, namely relation extraction (RE), plays a crucial role in a variety of knowledge-based applications, such as question answering (Yu et al., 2017) and large-scale knowledge graph construction. Previous methods (Zeng et al., 2014; Zeng et al., 2015; Xiao and Liu, 2016; Zhang et al., 2017; Zhang et al., 2018; Baldini Soares et al., 2019) focus on sentence-level RE, which predicts relations among entities in a single sentence. However, sentence-level RE models suffer from an inevitable limitation – they fail to recognize relations between entities across sentences. Hence, extracting relations at the document-level is necessary for a holistic understanding of knowledge in text.

There are several major challenges in effective relation extraction at the document-level. Firstly, the subject and object entities involved in a relation may appear in different sentences. Therefore a relation cannot be identified based solely on a single sentence. Secondly, the same entity may be mentioned multiple times in different sentences. Cross-sentence context information has to be aggregated to represent the entity better. Thirdly, the identification of many relations requires techniques of logical reasoning. This means these relations can only be successfully extracted when other entities and relations, usually spread across sentences, are identified implicitly or explicitly. As Figure 1 shows, it is easy to recognize the intra-sentence relations (*Maryland*, country, *U.S.*), (*Baltimore*, located in the administrative territorial entity, *Maryland*), and (*Eldersburg*, located in the administrative territorial entity, *Maryland*), since the subject and object appear in the same sentence. However, it is non-trivial to predict the inter-sentence relations between *Baltimore* and *U.S.*, as well as *Eldersburg* and *U.S.*, whose mentions do not appear in the same sentence and have long-distance dependencies. Besides, the identification of these two relation instances also

*Equal contribution.
†Corresponding author.

Error Type	Count
Intra-sentence	535
Inter-sentence	615
Logical Reasoning	242

Table 1: Statistics of bad cases in randomly sampled 100 documents from DocRED dev set for BiLSTM (Yao et al., 2019), with 1150 bad cases in total.

requires logical reasoning. For example, *Eldersburg* belongs to *U.S.* because *Eldersburg* is located in *Maryland*, which belongs to *U.S.*.

Recently, Yao et al. (2019) proposed a large-scale human-annotated document-level RE dataset, DocRED, to push sentence-level RE forward to document-level and it contains massive relation facts. Figure 1 shows an example from DocRED. We randomly sample 100 documents from the DocRED dev set and manually analyze the bad cases predicted by a BiLSTM-based model proposed by Yao et al. (2019). As shown in Table 1, the error type of inter-sentence and that of logical reasoning take up a large proportion of all bad cases, with 53.5% and 21.0% respectively. Therefore, in this paper, we aim to tackle these problems to extract relations from documents better.

Previous work in document-level RE do not consider reasoning (Gupta et al., 2019; Jia et al., 2019; Yao et al., 2019), or only use graph-based or hierarchical neural network to conduct reasoning in an implicit way (Peng et al., 2017; Sahu et al., 2019; Nan et al., 2020). In this paper, we propose a **G**raph **A**ggregation-and-**I**nference **N**etwork (GAIN) for document-level relation extraction. It is designed to tackle the challenges mentioned above directly. GAIN constructs a **h**eterogeneous **M**ention-level **G**raph (hMG) with two types of nodes, namely mention node and document node, and three different types of edges, i.e., intra-entity edge, inter-entity edge and document edge, to capture the context information of entities in the document. Then, we apply Graph Convolutional Network (Kipf and Welling, 2017) on hMG to get a document-aware representation for each mention. **E**ntity-level **G**raph (EG) is then constructed by merging mentions that refer to the same entity in hMG, on top of which we propose a novel path reasoning mechanism. This reasoning mechanism allows our model to infer multi-hop relations between entities.

In summary, our main contributions are as follows:

- We propose a novel method, Graph Aggregation-and-Inference Network (GAIN), which features a double graph design, to better cope with document-level RE task.
- We introduce a heterogeneous Mention-level Graph (hMG) with a graph-based neural network to model the interaction among different mentions across the document and offer document-aware mention representations.
- We introduce an Entity-level Graph (EG) and propose a novel path reasoning mechanism for relational reasoning among entities.

We evaluate GAIN on the public DocRED dataset. It significantly outperforms the previous state-of-the-art model by 2.85 F1 score. Further analysis demonstrates the capability of GAIN to aggregate document-aware context information and to infer logical relations over documents.

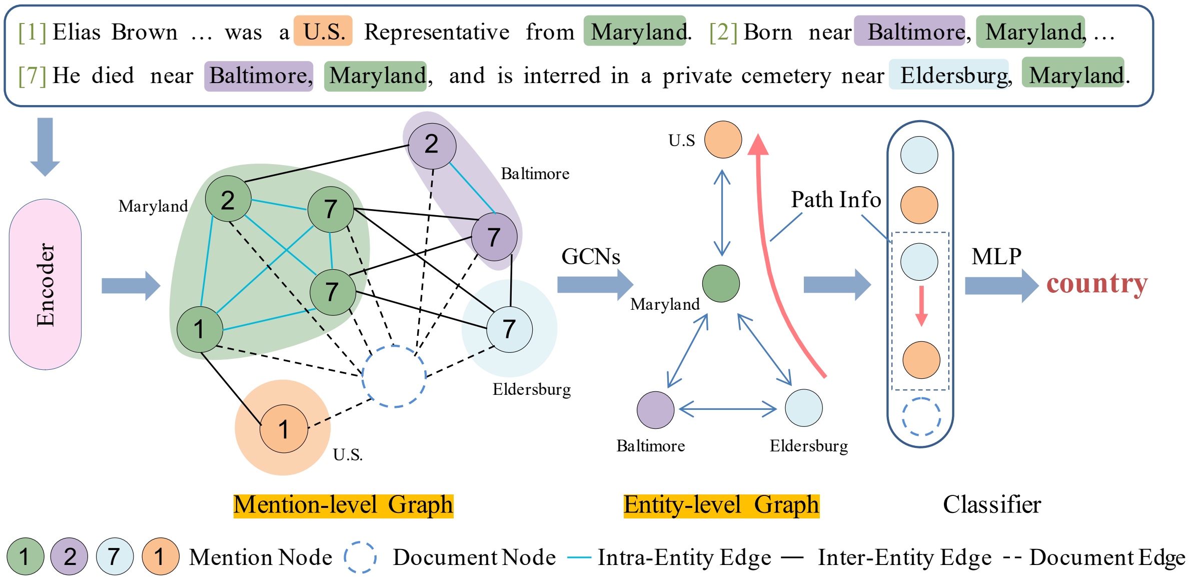
2 Task Formulation

We formulate the document-level relation extraction task as follows. Given a document comprised of N sentences $\mathcal{D} = \{s_i\}_{i=1}^{N}$ and a variety of entities $\mathcal{E} = \{e_i\}_{i=1}^{P}$, where $s_i = \{w_j\}_{j=1}^{M}$ refers to the i-th sentence consisting of M words, $e_i = \{m_j\}_{j=1}^{Q}$ and m_j refers to a span of words belonging to the j-th mention of the i-th entity, the task aims to extract the relations between different entities in $\mathcal{E}$, namely $\{(e_i, r_{ij}, e_j)|e_i, e_j \in \mathcal{E}, r_{ij} \in \mathcal{R}\}$, where $\mathcal{R}$ is a pre-defined relation type set.

In our paper, a relation r_{ij} between entity e_i and e_j is defined as inter-sentential, if and only if $S_{e_i} \cap S_{e_j} = \varnothing$, where S_{e_i} denotes those sentences containing mentions of e_i. Instead, a relation r_{ij} is defined as intra-sentential, if and only if $S_{e_i} \cap S_{e_j} \neq \varnothing$. We also define K-hop relational reasoning as predicting relation r_{ij} based on a K-length chain of existing relations, with e_i and e_j being the head and tail of the reasoning chain, i.e., $e_i \xrightarrow{r_1} e_m \xrightarrow{r_2} \dots e_n \xrightarrow{r_K} e_j \Rightarrow e_i \xrightarrow{r_{ij}} e_j$.

3 Graph Aggregation and Inference Network (GAIN)

GAIN mainly consists of 4 modules: encoding module (Sec. 3.1), mention-level graph aggregation module (Sec. 3.2), entity-level graph inference



[1] Elias Brown … was a U.S. Representative from Maryland. [2] Born near Baltimore, Maryland, …

[7] He died near Baltimore, Maryland, and is interred in a private cemetery near Eldersburg, Maryland.

Figure 2: The overall architecture of GAIN. First, A context encoder consumes the input document to get a contextualized representation of each word. Then, the Mention-level Graph is constructed with mention nodes and a document node. After applying GCN, the graph is transformed into Entity-level Graph, where the paths between entities are identified for reasoning. Finally, the classification module predicts target relations based on the above information. Different entities are in different colors. The number i in the mention node denotes that it belongs to the i-th sentence.

module (Sec. 3.3), classification module (Sec. 3.4), as is shown in Figure 2.

3.1 Encoding Module

In the encoding module, we convert a document $\mathcal{D} = \{w_i\}_{i=1}^n$ containing n words into a sequence of vectors $\{g_i\}_{i=1}^n$. Following Yao et al. (2019), for each word w_i in $\mathcal{D}$, we first concatenate its word embedding with entity type embedding and coreference embedding:

$$x_i = [E_w(w_i); E_t(t_i); E_c(c_i)] \tag{1}$$

where $E_w(\cdot)$, $E_t(\cdot)$ and $E_c(\cdot)$ denote the word embedding layer, entity type embedding layer and coreference embedding layer, respectively. t_i and c_i are named entity type and entity id. We introduce *None* entity type and id for those words not belonging to any entity.

Then the vectorized word representations are fed into an encoder to obtain the context sensitive representation for each word:

$$[g_1, g_2, \dots, g_n] = Encoder([x_1, x_2, \dots, x_n]) \tag{2}$$

where the $Encoder$ can be LSTM or other models.

3.2 Mention-level Graph Aggregation Module

To model the document-level information and interactions between mentions and entities, a heterogeneous Mention-level Graph (hMG) is constructed.

hMG has two different kinds of nodes: mention node and document node. Each mention node denotes one particular mention of an entity. And hMG also has one document node that aims to model the overall document information. We argue that this node could serve as a pivot to interact with different mentions and thus reduce the long distance among them in the document.

There are three types of edges in hMG:

- **Intra-Entity Edge:** Mentions referring to the same entity are fully connected with intra-entity edges. In this way, the interaction among different mentions of the same entity could be modeled.
- **Inter-Entity Edge:** Two mentions of different entities are connected with an inter-entity edge if they co-occur in a single sentence. In this way, interactions among entities could be modeled by co-occurrences of their mentions.
- **Document Edge:** All mentions are connected to the document node with the document edge.

With such connections, the document node can attend to all the mentions and enable interactions between document and mentions. Besides, the distance between two mention nodes is at most two with the document node as a pivot. Therefore long-distance dependency can be better modeled.

Next, we apply Graph Convolution Network (Kipf and Welling, 2017) on hMG to aggregate the features from neighbors. Given node u at the l-th layer, the graph convolutional operation can be defined as:

$$h_u^{(l+1)} = \sigma\left(\sum_{k\in\mathcal{K}} \sum_{v\in\mathcal{N}_k(u)} W_k^{(l)} h_v^{(l)} + b_k^{(l)}\right) \quad (3)$$

where $\mathcal{K}$ are different types of edges, $W_k^{(l)} \in \mathbb{R}^{d\times d}$ and $b_k^{(l)} \in \mathbb{R}^d$ are trainable parameters. $\mathcal{N}_k(u)$ denotes neighbors for node u connected in k-th type edge. σ is an activation function (e.g., ReLU).

Different layers of GCN express features of different abstract levels, and therefore in order to cover features of all levels, we concatenate hidden states of each layer to form the final representation of node u:

$$\mathbf{m}_u = [h_u^{(0)}; h_u^{(1)}; \ldots; h_u^{(N)}] \quad (4)$$

where $h_u^{(0)}$ is the initial representation of node u. For a mention ranging from the s-th word to the t-th word in the document, $h_u^{(0)} = \frac{1}{t-s+1}\sum_{j=s}^{t} g_j$ and for document node, it is initialized with the document representation output from the encoding module.

3.3 Entity-level Graph Inference Module

In this subsection, we introduce Entity-level Graph (EG) and path reasoning mechanism. First, mentions that refer to the same entity are merged to entity node so as to get the nodes in EG. Note that we do not consider document node in EG. For i-th entity node $\mathbf{e}_i$ mentioned N times, it is represented by the average of its N mention representations:

$$\mathbf{e}_i = \frac{1}{N}\sum_n \mathbf{m}_n \quad (5)$$

Then, we merge all inter-entity edges that connect mentions of the same two entities so as to get the edges in EG. The representation of directed edge from $\mathbf{e}_i$ to $\mathbf{e}_j$ in the EG is defined as :

$$\mathbf{e}_{ij} = \sigma\left(W_q[\mathbf{e}_i; \mathbf{e}_j] + b_q\right) \quad (6)$$

where W_q and b_q are trainable parameters, and σ is an activation function (e.g., ReLU).

Based on the vectorized edge representation, the i-th path between head entity $\mathbf{e}_h$ and tail entity $\mathbf{e}_t$ passing through entity $\mathbf{e}_o$ is represented as:

$$\mathbf{p}_{h,t}^i = [\mathbf{e}_{ho}; \mathbf{e}_{ot}; \mathbf{e}_{to}; \mathbf{e}_{oh}] \quad (7)$$

Note that we only consider two-hop paths here, while it can easily extend to multi-hop paths.

We also introduce attention mechanism (Bahdanau et al., 2015), using the entity pair $(\mathbf{e}_h, \mathbf{e}_t)$ as query, to fuse the information of different paths between $\mathbf{e}_h$ and $\mathbf{e}_t$.

$$s_i = \sigma([\mathbf{e}_h; \mathbf{e}_t] \cdot W_l \cdot \mathbf{p}_{h,t}^i) \quad (8)$$

$$\alpha_i = \frac{e^{s_i}}{\sum_j e^{s_j}} \quad (9)$$

$$\mathbf{p}_{h,t} = \sum_i \alpha_i \mathbf{p}_{h,t}^i \quad (10)$$

where α_i is the normalized attention weight for i-th path. Consequently, the model will pay more attention to useful paths. σ is an activation function.

With this module, an entity can be represented by fusing information from its mentions, which usually spread in multiple sentences. Moreover, potential reasoning clues are modeled by different paths between entities. Then they can be integrated with the attention mechanism so that we will take into account latent logical reasoning chains to predict relations.

3.4 Classification Module

For each entity pair $(\mathbf{e}_h, \mathbf{e}_t)$, we concatenate the following representations: (1) the head and tail entity representation $\mathbf{e}_h$ and $\mathbf{e}_t$ derived in the Entity-level Graph, with the comparing operation (Mou et al., 2016) to strengthen features, i.e., absolute value of subtraction between the representation of two entities, $|\mathbf{e}_h - \mathbf{e}_t|$, and element-wise multiplication, $\mathbf{e}_h \odot \mathbf{e}_t$; (2) the representation of document node in Mention-level Graph, $\mathbf{m}_{doc}$, as it can help aggregate cross-sentence information and provide document-aware representation; (3) the comprehensive inferential path information $\mathbf{p}_{h,t}$.

$$I_{h,t} = [\mathbf{e}_h; \mathbf{e}_t; |\mathbf{e}_h - \mathbf{e}_t|; \mathbf{e}_h \odot \mathbf{e}_t; \mathbf{m}_{doc}; \mathbf{p}_{h,t}] \quad (11)$$

Finally, we formulate the task as multi-label classification task and predict relations between entities:

$$P(r|\mathbf{e}_h, \mathbf{e}_t) = sigmoid\left(W_b\sigma(W_a I_{h,t} + b_a) + b_b\right) \quad (12)$$

where W_a, W_b, b_a, b_b are trainable parameters, σ is an activation function (e.g., ReLU). We use binary cross entropy as the classification loss to train our model in an end-to-end way:

$$\mathcal{L} = -\sum_{\mathcal{D}\in\mathcal{S}}\sum_{h\neq t}\sum_{r_i\in\mathcal{R}} \mathbb{I}\left(r_i = 1\right)\log P\left(r_i|\mathbf{e}_h, \mathbf{e}_t\right) + \mathbb{I}\left(r_i = 0\right)\log\left(1 - P\left(r_i|\mathbf{e}_h, \mathbf{e}_t\right)\right) \tag{13}$$

where $\mathcal{S}$ denotes the whole corpus, and $\mathbb{I}\left(\cdot\right)$ refers to indication function.

4 Experiments

4.1 Dataset

We evaluate our model on DocRED (Yao et al., 2019), a large-scale human-annotated dataset for document-level RE constructed from Wikipedia and Wikidata. DocRED has 96 relations types, $132,275$ entities, and $56,354$ relational facts in total. Documents in DocRED contain about 8 sentences on average, and more than 40.7% relation facts can only be extracted from multiple sentences. Moreover, 61.1% relation instances require various inference skills such as logical inference (Yao et al., 2019). we follow the standard split of the dataset, $3,053$ documents for training, $1,000$ for development and $1,000$ for test. For more detailed statistics about DocRED, we recommend readers to refer to the original paper (Yao et al., 2019).

4.2 Experimental Settings

In our GAIN implementation, we use 2 layers of GCN and set the dropout rate to 0.6, learning rate to 0.001. We train GAIN using AdamW (Loshchilov and Hutter, 2019) as optimizer with weight decay 0.0001 and implement GAIN under PyTorch (Paszke et al., 2017) and DGL (Wang et al., 2019b).

We implement three settings for our GAIN. **GAIN-GloVe** uses GloVe (100d) and BiLSTM (256d) as word embedding and encoder. **GAIN-BERT**$_{base}$ and **GAIN-BERT**$_{large}$ use BERT$_{base}$ and BERT$_{large}$ as encoder respectively and the learning rate is set to $1e^{-5}$.

4.3 Baselines and Evaluation Metrics

We use the following models as baselines.

Yao et al. (2019) proposed models to encode the document into a sequence of hidden state vector $\{\boldsymbol{h_i}\}_{i=1}^{n}$ using **CNN** (Fukushima, 1980), **LSTM** (Hochreiter and Schmidhuber, 1997), and **BiLSTM** (Schuster and Paliwal, 1997) as their encoder, and predict relations between entities with their representations. Other pre-trained models like **BERT** (Devlin et al., 2019), **RoBERTa** (Liu et al., 2019), and **CorefBERT** (Ye et al., 2020) are also used as encoder (Wang et al., 2019a; Ye et al., 2020) to document-level RE task.

Context-Aware, also proposed by Yao et al. (2019) on DocRED adapted from (Sorokin and Gurevych, 2017), uses an LSTM to encode the text, but further utilizes attention mechanism to absorb the context relational information for predicting.

BERT-Two-Step$_{base}$, proposed by Wang et al. (2019a) on DocRED. Though similar to BERT-RE$_{base}$, it first predicts whether two entities have a relationship and then predicts the specific target relation.

HIN-GloVe/HIN-BERT$_{base}$, proposed by Tang et al. (2020). Hierarchical Inference Network (HIN) aggregate information from entity-level, sentence-level, and document-level to predict target relations, and use GloVe (Pennington et al., 2014) or BERT$_{base}$ for word embedding.

LSR-GloVe/LSR-BERT$_{base}$, proposed by Nan et al. (2020) recently. They construct a graph based on the dependency tree and predict relations by latent structure induction and GCN. Nan et al. (2020) also adapted four graph-based state-of-the-art RE models to DocRED, including **GAT** (Velickovic et al., 2017), **GCNN** (Sahu et al., 2019), **EoG** (Christopoulou et al., 2019), and **AGGCN** (Guo et al., 2019). We also include their results.

Following Yao et al. (2019), we use the widely used metrics F1 and AUC in our experiment. We also use Ign F1 and Ign AUC, which calculate F1 and AUC excluding the common relation facts in the training and dev/test sets.

4.4 Results

We show GAIN's performance on the DocRED dataset in Table 2, in comparison with other baselines.

Among the models not using BERT or BERT variants, GAIN-GloVe consistently outperforms all sequential-based and graph-based strong baselines by $0.9 \sim 12.82$ F1 score on the test set. Among the models using BERT or BERT variants, GAIN-BERT$_{base}$ yields a great improvement of F1/Ign F1 on dev and test set by $2.22/6.71$ and $2.19/2.03$, respectively, in comparison with the strong baseline LSR-BERT$_{base}$. GAIN-BERT$_{large}$ also improves $2.85/2.63$ F1/Ign F1 on test set compared with

Model	Dev				Test	
	Ign F1	**Ign AUC**	**F1**	**AUC**	**Ign F1**	**F1**
CNN* (Yao et al., 2019)	41.58	36.85	43.45	39.39	40.33	42.26
LSTM* (Yao et al., 2019)	48.44	46.62	50.68	49.48	47.71	50.07
BiLSTM* (Yao et al., 2019)	48.87	47.61	50.94	50.26	48.78	51.06
Context-Aware* (Yao et al., 2019)	48.94	47.22	51.09	50.17	48.40	50.70
HIN-GloVe* (Tang et al., 2020)	51.06	-	52.95	-	51.15	53.30
GAT‡ (Velickovic et al., 2017)	45.17	-	51.44	-	47.36	49.51
GCNN‡ (Sahu et al., 2019)	46.22	-	51.52	-	49.59	51.62
EoG‡ (Christopoulou et al., 2019)	45.94	-	52.15	-	49.48	51.82
AGGCN‡ (Guo et al., 2019)	46.29	-	52.47	-	48.89	51.45
LSR-GloVe* (Nan et al., 2020)	48.82	-	55.17	-	52.15	54.18
GAIN-GloVe	**53.05**	**52.57**	**55.29**	**55.44**	**52.66**	**55.08**
BERT-RE$^*_{base}$ (Wang et al., 2019a)	-	-	54.16	-	-	53.20
RoBERTa-RE$^\dagger_{base}$	53.85	48.27	56.05	51.35	53.52	55.77
BERT-Two-Step$^*_{base}$ (Wang et al., 2019a)	-	-	54.42	-	-	53.92
HIN-BERT$^*_{base}$ (Tang et al., 2020)	54.29	-	56.31	-	53.70	55.60
CorefBERT-RE$^*_{base}$ (Ye et al., 2020)	55.32	-	57.51	-	54.54	56.96
LSR-BERT$^*_{base}$ (Nan et al., 2020)	52.43	-	59.00	-	56.97	59.05
GAIN-BERT$_{base}$	**59.14**	**57.76**	**61.22**	**60.96**	**59.00**	**61.24**
BERT-RE$^*_{large}$ (Ye et al., 2020)	56.67	-	58.83	-	56.47	58.69
CorefBERT-RE$^*_{large}$ (Ye et al., 2020)	56.73	-	58.88	-	56.48	58.70
RoBERTa-RE$^*_{large}$ (Ye et al., 2020)	57.14	-	59.22	-	57.51	59.62
CorefRoBERTa-RE$^*_{large}$ (Ye et al., 2020)	57.84	-	59.93	-	57.68	59.91
GAIN-BERT$_{large}$	**60.87**	**61.79**	**63.09**	**64.75**	**60.31**	**62.76**

Table 2: Performance on DocRED. Models above the first double line do not use pre-trained model. Results with * are reported in their original papers. Results with ‡ are performances of graph-based state-of-the-art RE models implemented in (Nan et al., 2020). Results with † are based on our implementation.

previous state-of-the-art method, CorefRoBERTa-RE$_{large}$. It suggests that GAIN is more effective in document-level RE tasks. We can also observe that LSR-BERT$_{base}$ improves F1 by 3.83 and 4.87 on dev and test set with GloVe embedding replaced with BERT$_{base}$. In comparison, our GAIN-BERT$_{base}$ yields an improvement by 5.93 and 6.16, which indicates GAIN can better utilize BERT representation.

4.5 Ablation Study

To further analyze GAIN, we also conduct ablation studies to illustrate the effectiveness of different modules and mechanisms in GAIN. We show the results of the ablation study in Table 3.

First, we remove the heterogeneous Mention-level Graph (hMG) of GAIN. In detail, we initialize an entity node in Entity-level Graph (EG) with Eq. 5 but replace $\mathbf{m}_n$ with $h_n^{(0)}$, and apply GCN to EG instead. Features in different layers of GCN are concatenated to obtain $\mathbf{e}_i$. Without hMG, the performance of GAIN-GloVe/GAIN-BERT$_{base}$ sharply drops by 2.08/2.02 Ign F1 score on dev set. This drop shows that hMG plays a vital role in capturing interactions among mentions belonging to the same and different entities and document-aware features.

Next, we remove the inference module. To be specific, the model abandon the path information between head and tail entity $\mathbf{p}_{h,t}$ obtained in Entity-level Graph, and predict relations only based on entity representation, $\mathbf{e}_h$ and $\mathbf{e}_t$, and document node representation, $\mathbf{m}_{doc}$. The inference module's removal results in poor performance across all metrics, for instance, 2.21/2.17 Ign F1 score decrease on the dev set for GAIN-GloVe/GAIN-BERT$_{base}$. It suggests that our path inference mechanism helps capture the potential K-hop inference paths to infer relations and, therefore, improve document-level RE performance.

Moreover, taking away the document node in hMG leads to 2.19/1.88 Ign F1 decrease on the dev set for GAIN-GloVe/GAIN-BERT$_{base}$. It helps GAIN aggregate the document information and works as a pivot to facilitate the information exchange among different mentions, especially those far away from each other within the document.

4.6 Analysis & Discussion

In this subsection, we further analyze both inter-sentential and inferential performance on the development set. The same as Nan et al. (2020), we report Intra-F1/Inter-F1 scores in Table 4, which only consider either intra- or inter-sentence relations respectively. Similarly, in order to evaluate the inference ability of the models, Infer-F1 scores are reported in Table 5, which only considers relations that engaged in the relational reasoning process . For example, we take into account the golden relation facts r_1, r_2, and r_3 if there exist $e_h \xrightarrow{r_1} e_o \xrightarrow{r_2} e_t$ and $e_h \xrightarrow{r_3} e_t$ when calculating

Model	Dev				Test	
	Ign F1	Ign AUC	F1	AUC	Ign F1	F1
GAIN-GloVe	**53.05**	**52.57**	**55.29**	**55.44**	**52.66**	**55.08**
- *hMG*	50.97	48.84	53.10	51.73	50.76	53.06
- *Inference Module*	50.84	48.68	53.02	51.58	50.32	52.66
- *Document Node*	50.86	48.68	53.01	52.46	50.32	52.67
GAIN-BERT$_{base}$	**59.14**	**57.76**	**61.22**	**60.96**	**59.00**	**61.24**
- *hMG*	57.12	51.54	59.17	54.61	57.31	59.56
- *Inference Module*	56.97	54.29	59.28	57.25	57.01	59.34
- *Document Node*	57.26	52.07	59.62	55.51	57.01	59.63

Table 3: Performance of GAIN with different embeddings and submodules.

Model	Intra-F1	Inter-F1
CNN*	51.87	37.58
LSTM*	56.57	41.47
BiLSTM*	57.05	43.49
Context-Aware*	56.74	42.26
LSR-GloVe*	60.83	48.35
GAIN-GloVe	**61.67**	**48.77**
- *hMG*	59.72	46.49
BERT-RE$^{*}_{base}$	61.61	47.15
RoBERTa-RE$_{base}$	65.65	50.09
BERT-Two-Step$^{*}_{base}$	61.80	47.28
LSR-BERT$^{*}_{base}$	65.26	52.05
GAIN-BERT$_{base}$	**67.10**	**53.90**
- *hMG*	66.15	51.42

Table 4: Intra- and Inter-F1 results on dev set of DocRED. Results with * are reported in (Nan et al., 2020).

Model	Infer-F1	P	R
CNN	37.11	32.81	42.72
LSTM	39.03	33.16	47.44
BiLSTM	38.73	31.60	50.01
Context-Aware	39.73	**33.97**	47.85
GAIN-GloVe	**40.82**	32.76	**54.14**
- *Inference Module*	39.76	32.26	51.80
BERT-RE$_{base}$	39.62	34.12	47.23
RoBERTa-RE$_{base}$	41.78	37.97	46.45
GAIN-BERT$_{base}$	**46.89**	**38.71**	**59.45**
- *Inference Module*	45.11	36.91	57.99

Table 5: Infer-F1 results on dev set of DocRED. P: Precision, R: Recall.

Infer-F1.

As Table 4 shows, GAIN outperforms other baselines not only in Intra-F1 but also Inter-F1, and the removal of hMG leads to a more considerable decrease in Inter-F1 than Intra-F1, which indicates our hMG do help interactions among mentions, especially those distributed in different sentences with long-distance dependency.

Besides, Table 5 suggests GAIN can better handle relational inference. For example, GAIN-BERT$_{base}$ improves 5.11 Infer-F1 compared with RoBERTa-RE$_{base}$. The inference module also plays an important role in capturing potential inference chains between entities, without which GAIN-BERT$_{base}$ would drop by 1.78 Infer-F1.

4.7 Case Study

Figure 3 also shows the case study of our proposed model GAIN, in comparison with other baselines. As is shown, BiLSTM can only identify two relations within the first sentence. Both BERT-RE$_{base}$ and GAIN-BERT$_{base}$ can successfully predict *Without Me* is part of *The Eminem Show*. But only GAIN-BERT$_{base}$ is able to deduce the performer and publication date of *Without Me* are the same as those of *The Eminem Show*, namely *Eminem* and *May 26, 2002*, where it requires logical inference across sentences.

5 Related Work

Previous approaches focus on sentence-level relation extraction (Zeng et al., 2014; Zeng et al., 2015; Wang et al., 2016; Zhou et al., 2016; Xiao and Liu, 2016; Zhang et al., 2017; Feng et al., 2018; Zhu et al., 2019). But sentence-level RE models face an inevitable restriction in practice, where many real-world relation facts can only be extracted across sentences. Therefore, many researchers gradually shift their attention into document-level relation extraction.

Several approaches (Quirk and Poon, 2017; Peng et al., 2017; Gupta et al., 2019; Song et al., 2018;

[1] ***The Eminem Show*** is the fourth studio album by American rapper ***Eminem***, released on ***May 26, 2002*** by Aftermath Entertainment, Shady Records, and Interscope Records.
[2] ***The Eminem Show*** includes the commercially successful singles "***Without Me***", "Cleanin' Out My Closet", "Superman", and "Sing for the Moment"....

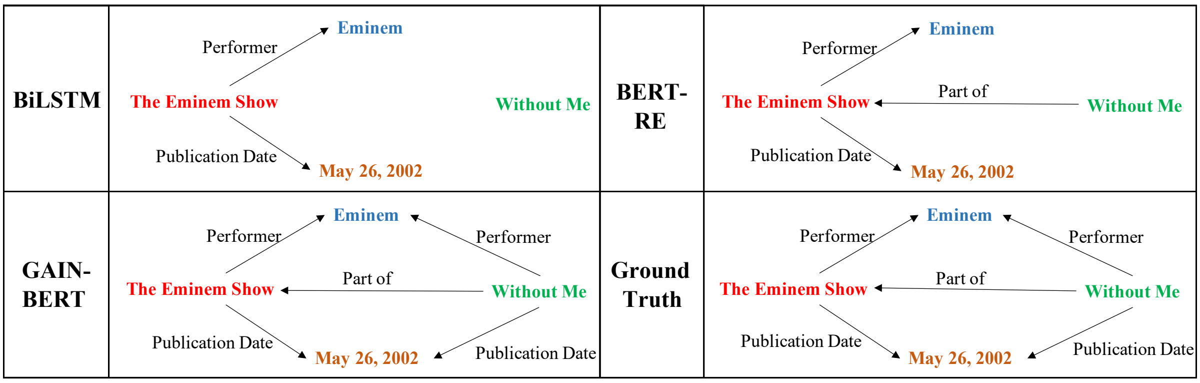


Figure 3: The case study of our proposed GAIN and baseline models. The models take the document as input and predict relations among different entities in different colors. We only show a part of entities within the documents and the according sentences due to the space limitation.

Jia et al., 2019) leverage dependency graph to better capture document-specific features, but they ignore ubiquitous relational inference in document. Recently, many models are proposed to address this problem. Tang et al. (2020) proposed a hierarchical inference network by considering information from entity-level, sentence-level, and document-level. However, it conducts relational inference implicitly based on a hierarchical network while we adopt the path reasoning mechanism, which is a more explicit way.

(Christopoulou et al., 2019) is one of the most powerful systems on document-level RE tasks recently. Compared to (Christopoulou et al., 2019) and other graph-based approaches to relation extraction, our architecture features many different designs with different motivations behind them. First, the ways of graph construction are different. We create two separate graphs of different levels to capture long-distance document-aware interactions and entity path inference information, respectively. While Christopoulou et al. (2019) put mentions and entities in the same graph. Moreover, they do not conduct graph node representation learning like GCN to aggregate interactive information on the constructed graph, only using the features from BiLSTMs to represent nodes. Second, the processes of path inference are different. Christopoulou et al. (2019) use a walk-based method to iteratively generate a path for every entity pair, which requires the extra overhead of hyper-parameter tuning to control the process of inference. Instead, we use an attention mechanism to selectively fuse all possible path information for the entity pair while without extra overhead.

When we were writing this paper, (Nan et al., 2020) make their work public as preprints, which adopt the dependency tree to capture the semantic information in the document. They put mention and entity nodes in the same graph and conduct inference implicitly by using GCN. Unlike their work, our GAIN presents mention node and entity node in different graphs to better conduct inter-sentence information aggregation and infer relations more explicitly.

Some other attempts (Verga et al., 2018; Sahu et al., 2019; Christopoulou et al., 2019) study document-level RE in a specific domain like biomedical RE. However, the datasets they use usually contain very limited relation types and entity types. For instance, CDR (Li et al., 2016) only has one type of relation and two types of entities, which may not be the ideal testbed for relational reasoning.

6 Conclusion

Extracting inter-sentence relations and conducting relational reasoning are challenging in document-level relation extraction.

In this paper, we introduce Graph Aggregation-and-Inference Network (GAIN) to better cope with document-level relation extraction, which features double graphs in different granularity. GAIN utilizes a heterogeneous Mention-level Graph to model the interaction among different mentions across the document and capture document-aware features. It also uses an Entity-level Graph with a proposed path reasoning mechanism to infer relations more explicitly.

Experimental results on the large-scale human-annotated dataset, DocRED, show GAIN outperforms previous methods, especially in inter-sentence and inferential relations scenarios. The ablation study also confirms the effectiveness of different modules in our model.

Acknowledgments

The authors would like to thank the anonymous reviewers for their thoughtful and constructive comments and ByteDance AI Lab for providing the computational resources for this work. This paper is supported in part by the National Key R&D Program of China under Grand No.2018AAA0102003, the National Science Foundation of China under Grant No.61876004 and 61936012.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

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

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pages 5779–5786.

Kunihiko Fukushima. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

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

Pankaj Gupta, Subburam Rajaram, Hinrich Schütze, and Thomas A. Runkler. 2019. Neural relation extraction within and across sentence boundaries. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pages 6513–6520.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level n-ary relation extraction with multiscale representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 3693–3704.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017*.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations, ICLR 2019*.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136.

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

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Chris Quirk and Hoifung Poon. 2017. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182.

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

M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph-state LSTM. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2226–2235.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1784–1789.

Hengzhu Tang, Yanan Cao, Zhenyu Zhang, Jiangxia Cao, Fang Fang, Shi Wang, and Pengfei Yin. 2020. HIN: hierarchical inference network for document-level relation extraction. In *Advances in Knowledge Discovery and Data Mining - 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11-14, 2020, Proceedings, Part I*, volume 12084 of *Lecture Notes in Computer Science*, pages 197–209.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. In *ICLR*.

Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 872–884.

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

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307.

Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander J Smola, and Zheng Zhang. 2019b. Deep graph library: Towards efficient and scalable deep learning on graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Minguang Xiao and Cong Liu. 2016. Semantic relation classification via hierarchical recurrent neural network with attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1254–1263.

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

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential reasoning learning for language representation. *arXiv preprint arXiv:2004.06870*, abs/2004.06870.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–581.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of*

COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 2335–2344.

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

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

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212.

Hao Zhu, Yankai Lin, Zhiyuan Liu, Jie Fu, Tat-Seng Chua, and Maosong Sun. 2019. Graph neural networks with generated parameters for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1331–1339.

A Hyperparameter settings

We use development set to manually tune the optimal hyperparameters for GAIN, based on the Ign F1 score. Hyperparameter settings for GAIN-GloVe, GAIN-BERT$_{base}$ and GAIN-BERT$_{large}$ are listed in Table 6, 7 and 8, respectively. The value of hyperparameters we finally adopted are in bold. Note that we do not tune all the hyperparameters.

Hyperparameter	Value
Batch Size	16, **32**
Learning Rate	**0.001**
Activation Function	**ReLU**, Tanh
Positive v.s. Negative Ratio	1, 0.5, **0.25**
Word Embedding Size	**100**
Entity Type Embedding Size	**20**
Coreference Embedding Size	**20**
Encoder Hidden Size	128, **256**
Dropout	0.2, **0.6**, 0.8
Layers of GCN	1, **2**, 3
GCN Hidden Size	**512**
Weight Decay	**0.0001**
Numbers of Parameters	63M
Hyperparameter Search Trials	12

Table 6: Settings for GAIN-GloVe.

Hyperparameter	Value
Batch Size	**5**
Learning Rate	**0.001**
Activation Function	**ReLU**, Tanh
Positive v.s. Negative Ratio	1, 0.5, **0.25**
Entity Type Embedding Size	**20**
Coreference Embedding Size	**20**
Dropout	0.2, **0.6**, 0.8
Layers of GCN	1, **2**, 3
GCN Hidden Size	**808**
Weight Decay	**0.0001**
Numbers of Parameters	217M
Hyperparameter Search Trials	20

Table 7: Settings for GAIN-BERT$_{base}$.

Hyperparameter	Value
Batch Size	**5**
Learning Rate	**0.001**
Activation Function	**ReLU**, Tanh
Positive v.s. Negative Ratio	1, 0.5, **0.25**
Entity Type Embedding Size	**20**
Coreference Embedding Size	**20**
Dropout	0.2, **0.6**, 0.8
Layers of GCN	1, **2**, 3
GCN Hidden Size	**1064**
Weight Decay	**0.0001**
Numbers of Parameters	512M
Hyperparameter Search Trials	20

Table 8: Settings for GAIN-BERT$_{large}$.