Dialogue Relation Extraction with Document-level Heterogeneous Graph Attention Networks

Hui Chen, Pengfei Hong, Wei Han, Navonil Majumder, Soujanya Poria

DeCLaRe Lab, Singapore University of Technology and Design, Singapore
hui_chen@mymail.sutd.edu.sg, {hongpengfei.emrys, henryhan88888}@gmail.com,
navo@nlp.cic.ipn.mx, sporia@sutd.edu.sg

Abstract

Dialogue relation extraction (DRE) aims to detect the relation between two entities mentioned in a multi-party dialogue. It plays an important role in constructing knowledge graphs from conversational data increasingly abundant on the internet and facilitating intelligent dialogue system development. The prior methods of DRE do not meaningfully leverage speaker information—they just prepend the utterances with the respective speaker names. Thus, they fail to model the crucial inter-speaker relations that may give additional context to relevant argument entities through pronouns and triggers. We, however, present a graph attention network-based method for DRE where a graph, that contains meaningfully connected speaker, entity, entity-type, and utterance nodes, is constructed. This graph is fed to a graph attention network for context propagation among relevant nodes, which effectively captures the dialogue context. We empirically show that this graph-based approach quite effectively captures the relations between different entity pairs in a dialogue as it outperforms the state-of-the-art approaches by a significant margin on the benchmark dataset DialogRE. Our code is released at: https://github.com/declare-lab/dialog-HGAT

1 Introduction

Relation extraction (RE) task aims to recognize relations between two entities present in a document. It plays a pivotal role in understanding unstructured text and constructing knowledge bases (Peng et al. 2017; Quirk and Poon 2017). Although the task of document-level relation extraction has been studied extensively in the past, the task of relation extraction from dialogues has yet to receive extensive study.

Conversational text exhibits intra- and inter-utterance relations (Poria et al. 2020), which makes it different from the text in previous document-level relation extraction. Most previous works focus on professional and formal literature like biomedical documents (Li et al. 2016; Wu et al. 2019) and Wikipedia articles (Elsahar et al. 2018; Yao et al. 2019; Mesquita et al. 2019). These kinds of datasets are well-formatted and logically coherent with clear referential semantics. Hence for most NLP tasks analyzing a few continuous sentences are enough to grasp pivotal information. However, for dialogue relation extraction, conversational text is sampled from daily chat, which is more casual in nature. Hence its logic is simpler but entangled and referential ambiguity always occurs to an external reader. Compared with formal literature, it has lower information density (Wang and Liu 2011) but is more difficult for model to understand. Moreover, compared with other document-level RE dataset such as DocRED, dialogue text has much more cross-sentence relations (Yu et al. 2020). Fig. 1 presents an example of dialogue relation extraction, taken from DialogRE (Yu et al. 2020) dataset. In order to infer the relation between *Speaker1* and *Emma*, we may need to find some triggers to recognize the characteristics of *Emma*. Triggers are evidences that can support the inference. As we can see, the following utterances are talking about *Emma*, and the key word *baby daughter* mentioned by *Speaker1* is a trigger, which provides an evidence that *Emma* is *Speaker1*'s daughter.

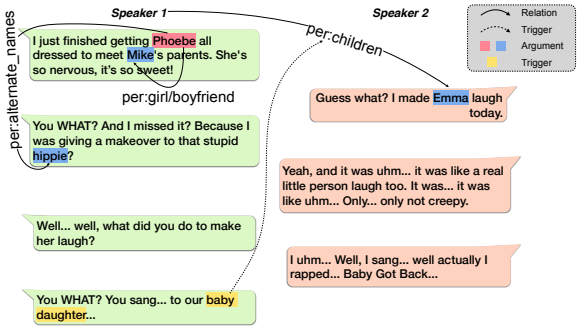

Figure 1: An example adapted from DialogRE dataset. Words with red and blue background represent subject and object entities. Words with yellow background represent triggers that facilitate the relation inference. Solid and dash lines stand for intra- and inter-utterance relations.

Prior works show that triggers of arguments facilitate the document-level relation inference. Thus, DocRED dataset (Yao et al. 2019) provides several supporting evidences for each argument pair. Some efforts utilize the dependency paths of arguments to find possible triggers. For example, LSR model (Nan et al. 2020) constructs meta dependency paths of each argument pair and aggregates all the word representations located in these paths to their model, in order to enhance model's reasoning ability. Sahu et al. (2019) uses syntactic parsing and coreference resolution to find intra- and inter-related words of each ar-

gument. Christopoulou, Miwa, and Ananiadou (2019) proposes an edge-oriented graph to synthesize argument-related information. These models are graph-based and have proven powerful in encoding long-distance information. However, for dialogue relation extraction, interlocutors exist in every utterance of the dialogue, and they are often considered as an argument. Although these previous approaches have utilized entity features of arguments, most of them employ meta dependency paths to find the related words, which results in the missing of necessary information related to speakers, since the speaker references have very little dependency features in each utterance. We think the structure of our graph allows it to model the intra- and inter-speaker relations through paths that involve conversational discourse and word-level semantics. This phenomenon enables the model to outshine the state-of-the-art frameworks int the task of dialogue level relation extraction.

In this work, we propose a simple yet effective attention-based heterogeneous graph neural network to tackle the dialogue relation extraction task by using multi-type features to create the graph and employing graph attention mechanism to propagate contextual information. Different from most of the previous works, our proposed model is customized for the relation extraction task in dialogue background, as we have specially modeled speaker information and designed a mechanism to propagate massages among different sentences for better inter-sentence representation learning. The remainder of this paper is organized as follows: Section 3 elaborates on our proposed framework; Section 4 introduces the used dataset and baseline models; Section 5 lays out the experiment results and analysis; Section 6 briefly discusses relevant works of heterogeneous graph neural networks; and Section 7 concludes the paper.

2 Background of Graph Attention Networks

2.1 Graph Attention Network (GAT)

The graph attention network is composed of graph attentional layers. Given a graph $G = (V, E)$ and its node embeddings $\mathbf{h} = \{\vec{h}_1, \vec{h}_2, ..., \vec{h}_N\}, \vec{h}_i \in \mathbb{R}^F$, GAT layer updates all the embeddings to $\mathbf{h}' = \{\vec{h}'_1, \vec{h}'_2, ..., \vec{h}'_n\}$ using a self-attention mechanism. In the attention computation part, each node only with its neighbour nodes serves as inputs to generate a set of attention weights, which is called masked attention:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}(\tilde{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j])\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}(\tilde{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k])\right)} \quad (1)$$

Where $\mathbf{W} \in \mathbb{R}^{F\times F'}$ is a weight matrix to linearly transform embeddings to another feature space, $\tilde{\mathbf{a}} \in \mathbb{R}^{2F'}$ weight matrix in a feed-forward neural network to generate the attention weights. Then these weights will be applied to original node features to generate new features:

$$\vec{h}'_i = \|_{k=1}^{K}\sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k\mathcal{W}^k\vec{h}_j\right) \quad (2)$$

2.2 Heterogeneous Graph Neural Network (HGNN)

Massive work on graph neural network treats the graph as homogeneous ones, where nodes and edges are of the same type. However, considering the complexity of the real world, the attributes of things and their relations vary greatly. As a result, it is difficult to use a homogeneous graph to describe them. Heterogeneous graphs, which assume multi-type nodes and edges, make mathematical modeling of the real world more approachable. A heterogeneous graph can be defined by a graph topology $G = (V, E)$ with a node type mapping: ϕ: $V \to \mathcal{A}$ and an edge type mapping ψ: $E \to \mathcal{R}$. Particularly, a graph is a heterogeneous graph when the types of nodes $|\mathcal{E}| > 1$ or the types of edges $|\mathcal{E}| > 1$.

To construct neural networks on heterogeneous graphs, effective information extraction and message passing scheme should be formulated. Meta-path (Sun et al. 2011), which has been used as a general structure to capture different semantics in heterogeneous graphs, is utilized in HGNNs. A meta-path is a path defined on the edge and node type set $\{\mathcal{R}, \mathcal{E}\}$, in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} ... \xrightarrow{R_l} A_{l+1}$. It specifies a composition relation $R = R_1 \circ ... \circ R_l$ between objects A_1 and A_{l+1}. In our work, we firstly build a heterogeneous graph composed of hierarchical and functional language components from the dataset, then applies heterogeneous graph attention operations on task-specified useful meta-paths to enhance the performance.

3 Methodology

3.1 Task Definition

Given a dialogue containing N utterances $\mathcal{D} = \{u_1, u_2, ..., u_N\}$ and argument pairs $\mathcal{A} = \{(x_1, y_1), (x_2, y_2), \dots\}$, where subject x_i and object y_i are entities mentioned in the dialogue, the goal is to identify the relation between argument pairs (x_i, y_i). For document-level relation extraction task, subject entity and object entity are often distributed in various sentences.

3.2 Model Overview

In this work, a conversation is represented as a heterogeneous graph for both intra- and inter-sentence relation inferences. We first utilize Utterance Encoder to encode sentential level utterance information. These utterance encodings along with word embeddings, speaker embeddings, argument embeddings, and entity-type embeddings are logically connected to form a heterogeneous graph—discussed in detail later in this section. Further, this graph is fed through a graph attention layer (Veličković et al. 2018) that aggregates information from the neighboring nodes.

Lastly, we concatenate the learned argument embeddings and feed them to a classifier. An overview of the proposed framework is shown in Fig. 2.

3.3 Data Preprocessing

We use spaCy[1] to tokenize utterances and at the same time, we obtain part-of-speech (POS) tags and named-entity types

[1]https://spacy.io

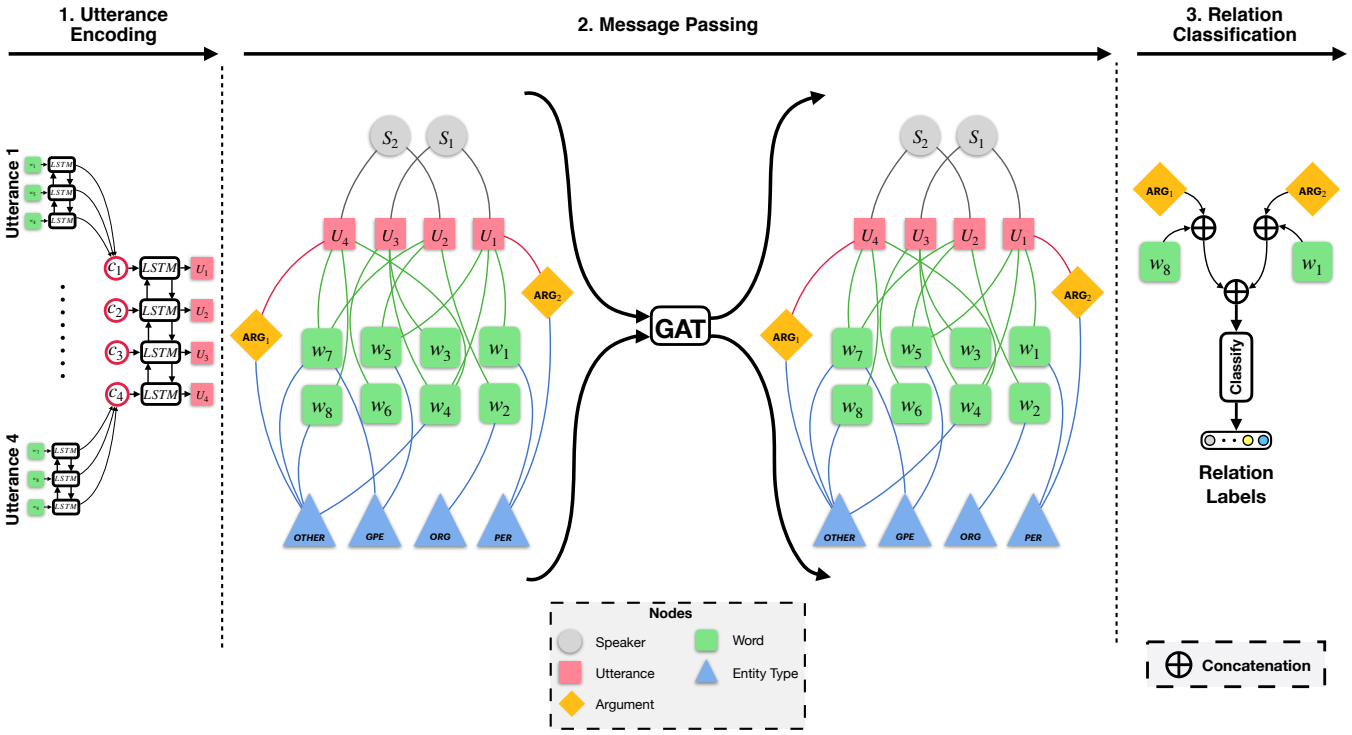

Figure 2: An overview of the proposed framework.

of each token.

3.4 Utterance Encoder

Given a dialogue $\mathcal{D}$, each GloVe (Pennington, Socher, and Manning 2014) initialized utterance u_i is first fed to a contextual encoder to obtain the contextualized representations of each constituent word. Bidirectional long short-term memory network (BiLSTM) is used as our contextual encoder. The operation of BiLSTM in our model can be defined as:

$$\overleftarrow{h_j^i} = LSTM_l(\overleftarrow{h_{j+1}^i}, e_j^i) \quad (3)$$

$$\overrightarrow{h_j^i} = LSTM_r(\overrightarrow{h_{j-1}^i}, e_j^i) \quad (4)$$

$$h_j^i = [\overleftarrow{h_j^i}; \overrightarrow{h_j^i}] \quad (5)$$

where $\overleftarrow{h_j^i}$ and $\overrightarrow{h_j^i}$ denote the hidden representations in the j-th layer of utterance u_i from two directions, h_j^i is the contextual representation which is the concatenation of $\overleftarrow{h_j^i}$ and $\overrightarrow{h_j^i}$, and e_j^i stands for the embedding of the j-th token in utterance u_i. Unlike the previous approaches (Christopoulou, Miwa, and Ananiadou 2019) that only rely on semantic-contextual features for utterance encoding, we also employ syntactic features. Thereby, we concatenate semantic word-embedding e_w initialized by GloVe (Pennington, Socher, and Manning 2014), syntactic Part of Speech embedding e_p, and entity-type embedding e_t to form token embedding e:

$$e = [e_w; e_p; e_t]. \quad (6)$$

To encode non-local contextual information between each utterances, we max pool the hidden states of each utterance-level BiLSTM (local LSTM), and then feed the sequence $c = \{c_1, c_2, ..., c_N\}$ to a conversational-level BiLSTM (global LSTM). The operation of global LSTM is the same as Eqs. (3) to (5).

3.5 Graph Construction

Node Construction There are five types of nodes in our proposed heterogeneous graph: utterance nodes, entity-type nodes, word nodes, speaker nodes, and argument nodes. Each type of node is used to encode a type of information in the dialogue.

Utterance and Entity-Type Nodes. Utterance nodes are used to represent the utterance-level information in a conversation. We use the outputs of utterance encoder which contains the utterance level encoding to initialize our utterance nodes.

Entity-type nodes represent the types of words that include a variety of named and numeric entities, such as PERSON or LOCATION. Naturally, each constituent word of a named entity is connected to its corresponding type node. Since the different mentions of the same entity may have different types in one conversation, we aim to capture all the type information of each entity via this entity-type node. For example, 'Frank' can be a string if it represents an alternative name, and at the same time, it can be a person if it refers to a speaker in the conversation. We believe that entity-type information has a positive influence on the relation-inference

process. Each Entity-Type node is initialized with our created Entity-Type embedding according to its entity type.

Word, Speaker, and Argument Nodes. Word nodes represent the vocabulary of the conversation. Each word node is connected with the utterance which contains the word and it is also connected with all the possible entity types that the word could have in the conversation. We initialize the states of word nodes with GloVe (Pennington, Socher, and Manning 2014).

Speaker node represents each unique speaker in the conversation. Each speaker node is connected with the utterances uttered by the speaker himself/herself. This type of node is initialized with its specific embedding and it is used to gather global information about the utterances made by the speaker.

Argument nodes are two special nodes that used to encode argument's relative positional information about the argument pair. One stands for the subject argument and the other represents the object argument. Similar to speaker nodes, argument nodes are also encoded by a specific embedding.

Edge Construction The proposed graph is undirected but the propagation has directions. There are five types of edges: utterance-word, utterance-argument, utterance-speaker, type-word, and type-argument. 'A-B' means there are edges between node A and B. Each edge has its own type. These edges are randomly initialized except utterance-word edge.

For the edge between utterance and word nodes, we adopt POS tags to initialize the edge features, since POS tags can reflect the local information of each word. This kind of edge aggregates not only global semantic features of the conversation but also local syntactic features to the word nodes.

Graph Attention Layer We use graph attention mechanism (Veličković et al. 2018) to aggregate discourse information and entity-type information to basic nodes. Let's take node i as an example. The graph attention mechanism described in the following shows how i's neighborhood j aggregates its information to i:

$$\mathcal{F}(h_i, h_j) = \text{LeakyReLU}(\mathbf{a}^T(\mathbf{W}_i h_i; \mathbf{W}_j h_j; \mathbf{E}_{ij}) \tag{7}$$

$$\alpha_{ij} = \text{softmax}(\mathcal{F}(h_i, h_j)) = \frac{\exp(\mathcal{F}(h_i, h_j))}{\sum_k \exp(\mathcal{F}(h_i, h_k))} \tag{8}$$

$$h'_i = ||_{k=1}^{K} \sigma(\sum_j \alpha_{ij}^k \mathbf{W}_q^k h_j) \tag{9}$$

where h_i and h_j are representations of node i and j, $\mathbf{W}_i$, $\mathbf{W}_j$, $\mathbf{W}_q$ and $\mathbf{a}^T$ are trainable weight matrices, $\mathbf{E}_{ij}$ is the edge weight matrix that is mapped to the multi-dimensional embedding space, α_{ij} is the attention weight between i and j, σ is an activation function, and $||$ is concatenation operation.

Message Propagation Although graph attention operation can effectively aggregate neighbor features, only one message passing will make the node structure relatively shallow. To make node features more informative, we update all basic nodes multiple times.

Our meta path is as follows: First, we use utterance nodes to update word nodes, speaker nodes and argument nodes; secondly, the updated word nodes propagate messages to entity-type nodes; then entity-type nodes update word nodes, argument nodes, and entity-type nodes; next we use word nodes, speaker nodes, and argument nodes to update utterance nodes; and lastly the updated utterance nodes update word nodes, speaker nodes and argument nodes. The path can be denoted as $V_u- > V_b- > V_t- > V_b- > V_u- > V_b$, where V_u, V_b, and V_t refer to utterance nodes, basic nodes, and entity-type nodes.

Following Wang et al. (2020), we add a residual connection (He et al. 2016) to avoid gradient vanishing during updating:

$$\hat{h_i} = \bar{h_i} + h'_i \tag{10}$$

where $\bar{h_i}$ is the output learned in the graph attention layer, and h'_i is the original input of the graph attention layer.

In our model, we first aggregate utterance nodes to basic nodes, so that semantic and syntactic features obtained in the utterance graph encoder can be intactly passed to word nodes, speaker nodes, and argument nodes. In message passing, except for graph attention operation, there is also a two-layer feed-forward network which can be denoted as:

$$h_i^{new} = \text{FFN}(\hat{h_i}) \tag{11}$$

Suppose we have the initial embeddings of utterance nodes, basic nodes and entity-type nodes, denoted as embedding matrices $\mathbf{H}_u = \{\mathbf{H}_u, \mathbf{H}_b, \mathbf{H}_t\}$, the message propagating process can be written as:

$$\mathbf{H}_b^1 = \text{GAT}(\mathbf{H}_b^0, \mathbf{H}_u^0) \tag{12}$$

$$\mathbf{H}_t^1 = \text{GAT}(\mathbf{H}_t^0, \mathbf{H}_b^1) \tag{13}$$

$$\mathbf{H}_b^2 = \text{GAT}(\mathbf{H}_b^1, \mathbf{H}_t^1) \tag{14}$$

$$\mathbf{H}_u^1 = \text{GAT}(\mathbf{H}_u^0, \mathbf{H}_b^2) \tag{15}$$

$$\mathbf{H}_b^3 = \text{GAT}(\mathbf{H}_b^2, \mathbf{H}_u^1) \tag{16}$$

where the GAT operation is the same as Eqs. (7) to (11). The superscripts represent it is the n^{th} new value of that matrix and 0 marks the initial value.

3.6 Relation Classifier

After the message propagation in the heterogeneous graph, we obtain new representations of all entities. We select the argument nodes τ_x and τ_y, as well as the corresponding word nodes e_x and e_y from basic nodes, and concatenate them. Finally, they are fed to a linear transformation and a sigmoid function to get the predictions:

$$e'_x = [\text{maxpool}(\tau_x); \text{maxpool}(e_x)] \tag{17}$$

$$e'_y = [\text{maxpool}(\tau_y); \text{maxpool}(e_y)] \tag{18}$$

$$e' = [e'_x; e'_y] \tag{19}$$

$$P(r|e_x, e_y) = \sigma(\mathbf{W}_e e' + b_e)_r \tag{20}$$

where $P(r|e_x, e_y)$ is the probability of relation type r given argument pair (e_x, e_y), $\mathbf{W}_e$ and b_e are linear transformation weight and bias vector, $maxpool$ is max pooling operation, and σ is sigmoid function.

4 Experiments

4.1 Dataset Used

We evaluate the proposed framework on the DialogRE dataset (Yu et al. 2020), which contains totally 1,788 dialogues and 10,168 relational triples. DialogRE is adapted from the complete transcripts of *Friends*, which is a widely used corpus in dialogue research these years (Chen, Zhou, and Choi 2017; Zhou and Choi 2018; Yang and Choi 2019), and there are 36 possible relation types, most of which focus on biographical attributes of person entities. Each dialogue contains several relational triples (x, y, r), and the task is to predict the relation r between each entity pair (x, y). In the experiments, the dataset is partitioned into train, dev, and test set with roughly 60/20/20 ratio. Following the evaluation metrics of DialogRE, we report macro F1 scores of the proposed model and all the baselines in both the standard and conversational settings.

4.2 Settings and Hyperparameters

In our experiments, we tune the parameters of batch size, learning rate, and BiLSTM hidden size by testing the performance on the validation set. Table 1 lists the major parameters used in our experiments.

Parameter	Value
Word Embedding Dimension	300
NER Embedding Dimension	30
POS Embedding Dimension	30
Local BiLSTM Hidden Size	200
Local BiLSTM Layers	2
Global BiLSTM Hidden Size	128
Global BiLSTM Layers	2
Multihead Attention Number	10
Learning Rate	0.0005
Batch Size	16
Edge Embedding Dimension	50

Table 1: Parameter settings.

4.3 Baseline models

Sequence-based Models We select convolutional neural networks(CNN) (Zeng et al. 2014), LSTM, and BiLSTM (Cai, Zhang, and Wang 2016) as the sequence-based baselines. These models take word embeddings, mention embeddings, and type embeddings as features. Concretely, they use GloVe and spaCy to get word embeddings and label named-entity types, and then take an average of all the embeddings of mention names for each entity to get mention embeddings.

Graph-based Models As our proposed model is graph-based, we also select two graph-based models AGGCN (Guo, Zhang, and Lu 2019) and LSR (Nan et al. 2020) as the baselines. AGGCN directly feeds the full dependency tree of each sentence to a graph convolutional network which takes self-attention weights as soft edges. It achieves state-of-the-art results in various relation extraction tasks. LSR adopts an adaptation of Kirchhoffs Matrix-Tree Theorem (Tutte 1984; Koo et al. 2007) to induce the latent dependency structure of each document and then feeds the latent structure to a densely connected graph convolutional network to inference the relations. These graph-based models both utilize dependency information to construct the inference graph.

5 Results and Analysis

5.1 Comparision with Baselines

We present our main results on DialogRE dataset in Table 2. As shown in Table 2, our model surpasses the state-of-the-art method by 9.6%/7.5% F1 scores, and 8.4%/5.7% $F1_c$ scores in both validation and test sets, which demonstrates the effectiveness of the information propagation along task-specific functional meta-paths in the heterogeneous graph. Whatever purely sequential models or graph-based models that are built from local transformers focus on modeling the sequence within a sentence scope. As a result, inter-sentence communication usually passes through a long distance, which causes information loss or disruption. However, this kind of information exchange is critically important for dialog-style text, because logical connections are not locally compact within adjacent sentences, instead they are spread over the whole conversations. Our proposed model, on the opposite, constructs a heterogeneous graph with shorter distances between logically closed but syntactically faraway word pairs. Hence the long-distance issue is mitigated.

We also compare the model sizes as an efficiency indicator. Although creating numerous nodes and edges inevitably brings overhead, the total number of parameters is still moderate.

5.2 Ablation Study

To understand the impact of the components in our model, we perform ablation study using our proposed model on DialogRE dataset. The ablation results are shown in Table 3. First, we remove local LSTM and global LSTM. The results showing drops in all the evaluation metrics prove that the contextual encoder plays an important role in semantic feature extraction. Second, we remove the specific argument nodes and have observed that F1 and $F1_c$ scores decrease to 55.0% and 50.2% on test set. This proves our design on argument nodes effectively synthesize argument features to the model. Further, we test the performance of the syntactic features we inject by removing POS embedding, NER embedding, and POS edge features. All the scores decrease, and specifically, the removal of POS embedding leads to about 2% drops in all the evaluation metrics.

Model	#params	Dev		Test	
		$F1$	$F1_c$	$F1$	$F1_c$
Majority (Yu et al. 2020)	-	38.9	38.7	35.8	35.8
CNN (Yu et al. 2020)	-	46.1	43.7	48.0	45.0
LSTM (Yu et al. 2020)	-	46.7	44.2	47.4	44.9
BiLSTM (Yu et al. 2020)	4.1M	48.1	44.3	48.6	45.0
AGGCN (Guo, Zhang, and Lu 2019)	3.7M	46.6	40.5	46.2	39.5
LSR (Nan et al. 2020)	20.5M	44.5	-	44.4	-
DHGAT(Ours)	4.0M	**57.7**	**52.7**	**56.1**	**50.7**

Table 2: Main results on DialogRE dataset. Values in the #params column refer to parameter sizes of the models. $F1$ and $F1_c$ are macro F1 scores under standard setting and conversational setting, respectively. The unit of all the scores is %.

Rand init vs. GloVe Additionally, we have compared different initialization strategies on word nodes. When we transfer GloVe initialization method to a random but trainable initialization method, we can observe a 3% to 4% decrease in all the metrics. This demonstrates our GloVe initialization strategy retains word features which have a positive influence on the performance.

Is our design of meta path optimal? We test the performance of our message propagation strategy via changing the update strategies. In our proposed model, those basic nodes composed of word, speaker, and argument nodes are updated totally thrice, i.e., they are first updated by utterance nodes, second updated by entity type nodes, and ultimately updated by utterance again. In our ablation study, we try to update basic nodes once, which means basic nodes are only updated by utterance nodes. Results present a dramatically drop of the evaluation scores, especially the standard F1 scores. However, if we add two more updates, that is to say, after our default updates, entity type nodes update basic nodes again and then utterance nodes update basic again, the results don't have an increase on the evaluation scores but decrease a bit. This proves that our message propagation strategy is the optimum now. If we only update the basic nodes once, node features are not informative enough. But if we update the basic nodes too many times, the features may be overfitted.

5.3 Case Studies

In the dataset, 95% of relation pairs have argument pairs that span two sentences. Therefore, it is crucial to model long range inter-sentential relationships. Our model can propagate relational information more effectively. Comparing to the LSTM model, speaker nodes, utterance nodes and unique word nodes shorten the information propagation path between two argument nodes. Considering the following example in Fig. 3. subject a - 'Mindy' and object b - 'Speaker 1' have relationship 'per:friends' indicated by the trigger 'my best friend' in the first utterance. The entity information is relayed from 'Mindy' to 'Speaker 1' in the update process: 'speaker 1' node aggregate utterance level information from its neighbor nodes that contains a. the relation trigger 'best friend'. b. utterance level information that contains the subject 'Mindy'. However, for bi-LSTM model, it will need to overcome long range of irrelevant information that will affect the final performance.

Model	Dev		Test	
	$F1$	$F1_c$	$F1$	$F1_c$
Full model	57.7	52.7	56.1	50.7
− Local BiLSTM	54.9	50.0	55.3	50.3
− Global BiLSTM	54.7	50.2	53.5	48.7
− Argument Nodes	56.0	51.3	55.0	50.2
− POS Embedding	54.6	50.9	53.0	48.5
− NER Embedding	56.8	51.5	54.2	49.2
− POSInitEdge	56.9	52.4	54.7	50.4
− Random Init Word Nodes	53.6	48.3	52.8	47.5
− Update Basic Nodes(t=1)	47.5	45.3	47.6	44.6
− Update Basic Nodes(t=5)	55.2	50.6	54.6	49.2

Table 3: Ablation results on DialogRE dataset. 't' means the number of updates for basic nodes. The unit of all the scores is %.

5.4 Error Analysis

Entity-type information involves in the information propagation process and thus affect the contents of output embeddings. The model is prone to make incorrectly biased predictions which highly relies on the entity types of two arguments if it fails to acquire enough certainty from other information sources. For example, given an entity pair of two human names, both with the named entity type 'PERSON'. Sometimes the model inclines to deem the relationship between the pair to be 'per:alternate_name' instead of correct 'per:alumni' or 'per:roommate'. This is because for all of these classes, 'PERSON-PERSON' is a preferable entity-type pair. However, the class 'per:alternate_name' (22.01%) presents more frequently than 'per:alumni' (1.83%) and 'per:roommate' (1.29%) in the dataset. When information aggregated from all sources other than entity pair is not evident for judgment, entity bias misguides the model to wrong classification results.

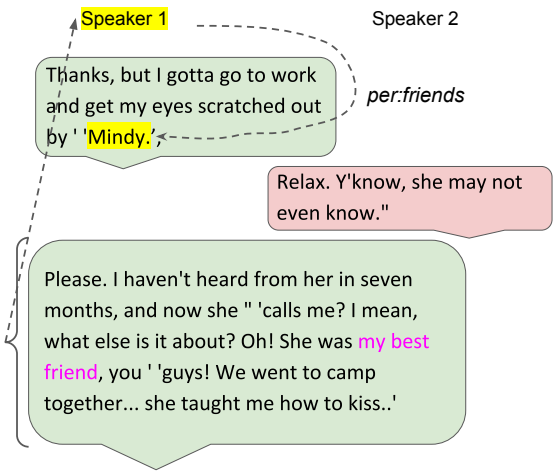

Figure 3: Case study: an example to show the effective message propagation between argument pairs

6 Related Work

Graph-based models have raised popular attention from NLP researchers, as it is demonstrated as a powerful mathematical tool to represent complicated syntactic and semantic relations among structured language data. Early work applies classic graph processing algorithms onto language graphs. Pang and Lee (2004) construct a text graph and adopt the minimum-cut method to cluster the nodes for sentiment analysis. Agirre and Soroa (2009) leverage PageRank algorithm on personalized subgraphs of a wordnet to disambiguate polysemous words according to connected context words.

Recently, with the achievement of graph neural networks (Kipf and Welling 2017), to incorporate syntactic features, which are easy to be expressed by graphs, into end-to-end learning models becomes a growing trend. Peng et al. (2017) firstly try to build a computation graph from syntactic parsing trees and employing graph LSTM to obtain better word embeddings for multi-ary relation extraction. Zhang, Qi, and Manning (2018) design a pruning algorithm for syntactic graphs and add a graph convolution layer on top of the sequential LSTM encoder in the learning process. The combination with typical attention-based language models such as transformer (Vaswani et al. 2017) is also studied. The work in (Cai and Lam 2020; Yao, Wang, and Wan 2020) use transformer-based graph convolutional networks to explicitly encode relations among distant syntactic nodes, to address the long-distance propagation issue.

Other works introduce heterogeneous graph neural networks into NLP tasks, like text classification (Linmei et al. 2019), text summarization (Wang et al. 2020), user profiling (Chen et al. 2019), and event categorization (Peng et al. 2019). These works prove that heterogeneous graph neural network is a powerful tool in NLP. For the relation extraction task, Christopoulou, Miwa, and Ananiadou (2019) construct an edge-oriented heterogeneous graph that contains sentence, mention, and entity information. However, syntactic information is neglected in their model. Different from them, homogeneous nodes in our graph is all independent, and we take syntactic features to initialize sentence information as well as edges features.

7 Conclusion

In this work, we present an attention-based heterogeneous graph to deal with the dialogue relation extraction task. This heterogeneous graph attention network has modeled multi-type features of the conversation, like utterance, word, speaker, argument, and entity type information. On the benchmark DialogRE dataset, our proposed framework outperforms the strong baselines and the state-of-the-art approaches by a significant margin, which proves the proposed framework can effectively capture relations between different entities in the conversation. Future works will focus on applying the relation knowledge to assist dialogue generation.

References

Agirre, E.; and Soroa, A. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 33–41.

Cai, D.; and Lam, W. 2020. Graph Transformer for Graph-to-Sequence Learning. In *AAAI*, 7464–7471.

Cai, R.; Zhang, X.; and Wang, H. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 756–765.

Chen, H. Y.; Zhou, E.; and Choi, J. D. 2017. Robust Coreference Resolution and Entity Linking on Dialogues: Character Identification on TV Show Transcripts. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, 216–225. Vancouver, Canada: Association for Computational Linguistics.

Chen, W.; Gu, Y.; Ren, Z.; He, X.; Xie, H.; Guo, T.; Yin, D.; and Zhang, Y. 2019. Semi-supervised User Profiling with Heterogeneous Graph Attention Networks. In *IJCAI*, volume 19, 2116–2122.

Christopoulou, F.; Miwa, M.; and Ananiadou, S. 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)*, 4927–4938.

Elsahar, H.; Vougiouklis, P.; Remaci, A.; Gravier, C.; Hare, J.; Laforest, F.; and Simperl, E. 2018. T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Guo, Z.; Zhang, Y.; and Lu, W. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 241–251.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

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

Koo, T.; Globerson, A.; Carreras, X.; and Collins, M. 2007. Structured prediction models via the matrix-tree theorem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 141–150.

Li, J.; Sun, Y.; Johnson, R. J.; Sciaky, D.; Wei, C.-H.; Leaman, R.; Davis, A. P.; Mattingly, C. J.; Wiegers, T. C.; and Lu, Z. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database* 2016.

Linmei, H.; Yang, T.; Shi, C.; Ji, H.; and Li, X. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. 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)*, 4823–4832.

Mesquita, F.; Cannaviccio, M.; Schmidek, J.; Mirza, P.; and Barbosa, D. 2019. Knowledgenet: A benchmark dataset for knowledge base population. 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)*, 749–758.

Nan, G.; Guo, Z.; Sekulic, I.; and Lu, W. 2020. Reasoning with Latent Structure Refinement for Document-Level Relation Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1546–1557.

Pang, B.; and Lee, L. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 271–278.

Peng, H.; Li, J.; Gong, Q.; Song, Y.; Ning, Y.; Lai, K.; and Yu, P. S. 2019. Fine-grained event categorization with heterogeneous graph convolutional networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 3238–3245. AAAI Press.

Peng, N.; Poon, H.; Quirk, C.; Toutanova, K.; and Yih, W.-t. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics* 5: 101–115.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Poria, S.; Hazarika, D.; Majumder, N.; and Mihalcea, R. 2020. Beneath the Tip of the Iceberg: Current Challenges and New Directions in Sentiment Analysis Research. *arXiv preprint arXiv:2005.00357* .

Quirk, C.; and Poon, H. 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*, 1171–1182.

Sahu, S. K.; Christopoulou, F.; Miwa, M.; and Ananiadou, S. 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*, 4309–4316.

Sun, Y.; Han, J.; Yan, X.; Yu, P. S.; and Wu, T. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment* 4(11): 992–1003.

Tutte, W. T. 1984. Graph Theory. In *Claren- don Press*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *6th International Conference on Learning Representations*.

Wang, D.; Liu, P.; Zheng, Y.; Qiu, X.; and Huang, X. 2020. Heterogeneous Graph Neural Networks for Extractive Document Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6209–6219.

Wang, D.; and Liu, Y. 2011. A pilot study of opinion summarization in conversations. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human language technologies*, 331–339.

Wu, Y.; Luo, R.; Leung, H. C.; Ting, H.-F.; and Lam, T.-W. 2019. Renet: A deep learning approach for extracting gene-disease associations from literature. In *International Conference on Research in Computational Molecular Biology*, 272–284. Springer.

Yang, Z.; and Choi, J. D. 2019. FriendsQA: Open-domain question answering on TV show transcripts. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, 188–197.

Yao, S.; Wang, T.; and Wan, X. 2020. Heterogeneous Graph Transformer for Graph-to-Sequence Learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7145–7154.

Yao, Y.; Ye, D.; Li, P.; Han, X.; Lin, Y.; Liu, Z.; Liu, Z.; Huang, L.; Zhou, J.; and Sun, M. 2019. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 764–777.

Yu, D.; Sun, K.; Cardie, C.; and Yu, D. 2020. Dialogue-Based Relation Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4927–4940.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2335–2344.

Zhang, Y.; Qi, P.; and Manning, C. D. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2205–2215.

Zhou, E.; and Choi, J. D. 2018. They exist! introducing plural mentions to coreference resolution and entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, 24–34.