Dialogue Relation Extraction with Document-Level Heterogeneous Graph Attention Networks

Hui Chen1 · Pengfei Hong1 · Wei Han1 · Navonil Majumder1 · Soujanya Poria1

Received: 21 July 2022 / Accepted: 2 January 2023 / Published online: 20 January 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
We propose a heterogeneous graph attention network to address the problem of dialogue relation extraction. Compared with several popular sequence-based and graph-based models, our method shows superior performance on the benchmark dataset DialogRE. The implementation of this work can be found at https://github.com/declare-lab/dialog-HGAT Dialogue relation extraction aims to detect the relation between pairs of entities mentioned in a multi-party dialogue. It plays an essential role in understanding the deep logic of dialogues and facilitating the development of intelligent dialogue systems. We introduce a heterogeneous graph attention network to model the cross-sentence relations in a conversation. This heterogeneous graph attention network has modeled multi-type features of the conversation, such as utterance, word, speaker, argument, and entity type information. We compare our method with several popular baselines such as convolutional neural networks and long short-term memory, experimental results show our model outperforms the state-of-the-art method by 9.4%/7.8% F1 scores, and 6.6%/3.9% F1c scores in both validation and test sets with only 4.0M parameters. In this work, we present an attention-based heterogeneous graph network to deal with the dialogue relation extraction task in an inductive manner. Experimental results on the dataset DialogRE confirm the effectiveness of our method.

Keywords Dialogue relation extraction · Heterogeneous graph neural networks · Cross-sentence relations

Introduction
The relation extraction (RE) task aims to identify relations between pairs of entities that exist in a document. It plays a pivotal role in understanding unstructured text and constructing knowledge bases [1, 2]. 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.

Most previous works in this field focus on the professional and formal literature like biomedical documents [3, 4] and Wikipedia articles [5–8]. These kinds of datasets are well-formatted and logically coherent with clear referential semantics. Hence, for most NLP tasks, analyzing a few continuous sentences is enough to grasp pivotal information. However, in dialogue relation extraction, conversational text is sampled from daily chat, which is more casual. Hence its logic is simpler but more entangled, and the referential ambiguity always occurs to an external reader. Compared with formal literature, it has lower information density [9] and thus is more difficult for models to understand. Moreover, compared with other document-level RE datasets such as DocRED, dialogue text has much more cross-sentence relations [10].

Figure 1 presents an example of the target dialogue, taken from DialogRE [10] 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 shreds of evidence that can support the inference. As we can see, the following utterances are talking about Emma, and the keyword *baby daughter* mentioned by Speaker1 is
a trigger, which provides evidence that Emma is Speaker1’s daughter.

Prior works show that triggers of arguments facilitate the document-level relation inference. Thus, the DocRED dataset [6] provides several supporting pieces of evidence for each argument pair. Some efforts utilize the dependency paths of arguments to find possible triggers. For example, LSR model [11] constructs meta dependency paths of each argument pair and aggregates all the word representations located in these paths to their model to enhance the model’s reasoning ability. Sahu et al. [12] uses syntactic parsing and coreference resolution to find intra- and inter-related words of each argument. Christopoulou et al. [13] 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 neglect necessary information related to speakers, since the speaker references have very few dependency features in each utterance. In this work, we formulate the dialogue relation extraction task as a classification problem, where we design a graph attention network to model semantic, syntactic, and speaker information. Compared with other graph-based models in the relation extraction task, our model is lightweight, without any costly matrix operation, and can generalize to completely unseen graphs.

In this paper, we propose a simple yet effective attention-based heterogeneous graph neural network to tackle the dialogue relation extraction task in an inductive manner. We use multi-type features to create the graph and employ a 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 messages among different sentences for better inter-sentence representation learning.

The remainder of this paper is organized as follows: the “Related Work” section briefly discusses relevant works of heterogeneous graph neural networks; the “Method” section elaborates on our proposed framework; the “Experiments” section introduces the used dataset and baseline models; the “Results and Analysis” section lays out the experiment results and analysis; the “Conclusion” section concludes the paper.

**Related Work**

Graph-based models have raised widespread 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 to language graphs. Pang and Lee [14] constructed a text graph and adopt the minimum-cut method to cluster the nodes for sentiment analysis. Agirre and Soroa [15] leveraged the PageRank algorithm on personalized subgraphs of wordnet to disambiguate polysemous words according to connected context words.

Recently, graph neural networks (GNN) [16] becomes popular in relation extraction tasks. For example, Peng et al. [1] tried to build a computation graph from syntactic parsing trees and employed graph LSTM to obtain better word embeddings for relation extraction. Zhang et al. [17] designed 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 [18] is also studied. Cai and Lam [19] and Yao et al. [20] used transformer-based graph convolutional networks to explicitly encode relations among distant syntactic nodes, to address the long-distance propagation issue.

Based on GNN, heterogeneous graph neural networks are proposed and have been applied in many NLP tasks, like text classification [21], text summarization [22], user profiling [23], and event categorization [24]. The prior work proves that a heterogeneous graph neural network is a powerful tool in NLP. For the relation extraction task, Christopoulou et al. [13] constructed an edge-oriented heterogeneous graph that contains the sentence, mention, and entity information. However, syntactic information is neglected in their model. Differently, homogeneous nodes in our graph are all independent, and we take syntactic features to initialize sentence information as well as edge features.

Method

Task Definition

Given a dialogue containing $N$ utterances $D = \{u_1, u_2, ..., u_N\}$ and a couple of argument pairs $A = \{(x_1, y_1), (x_2, y_2), \ldots\}$, 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 the document-level relation extraction task, there are many cross-sentence relations supported by various sentences.

Model Overview

In this work, we introduce an attention-based graph network to tackle the problem where each conversation is represented as a heterogeneous graph. We first utilize an utterance encoder, which is composed of two bidirectional long short-term memory (BiLSTM) networks to encode conversational information. These utterance encodings, along with word embeddings, speaker embeddings, argument embeddings, and named entity type embedding, are logically connected to form a heterogeneous graph, which will be discussed in detail later in this section. Further, this graph is fed through five graph attention layers [25] that aggregate information from the neighboring nodes. Lastly, we concatenate the learned argument embeddings and feed them to a classifier. An overview of the proposed model is shown in Fig. 2.

Feature Extraction

In the data preprocessing period, we use spaCy\(^1\) to tokenize utterances, and at the same time, we obtain part-of-speech (POS) tags as well as named entity types of each token.

Utterance Encoder

Given a dialogue $D = \{u_1, u_2, ..., u_N\}$, we use GloVe [26] to initialize the word embeddings and then feed them to a contextual Bidirectional Long-Short-Term Memory network (BiLSTM) to obtain contextualized representations. The operation of BiLSTM can be defined as:

\[
\overline{R}_j = \text{LSTM}_l(h_{j+1}^l, e_j)
\]

\[
\overline{R}_j = \text{LSTM}_r(h_{j-1}^r, e_j)
\]

\(^1\)https://spacy.io
where $\overrightarrow{h_j}$ and $\overleftarrow{h_j}$ denote the hidden representations in the $j$-th layer of utterance $u_j$ from two directions, $h_j^i$ is the contextual representation which is the concatenation of $\overrightarrow{h_j}$ and $\overleftarrow{h_j}$, and $e_j^i$ stands for the embedding of the $j$-th token in utterance $u_j$.

Unlike the previous approaches [11, 13] that only adopt semantic contextual features in utterance encoding, we add syntactic features such as POS tags and named entity types to the contextual representations. The embedding of each token in the utterance can be described as:

$$ e = [e_w, e_p, e_t] $$

where we concatenate word embedding $e_w$ initialized by GloVe [26], syntactic POS embedding $e_p$, and named entity type embedding $e_t$ to form the token embedding $e$.

Moreover, we believe conversation-level contextual features play an important role in understanding a conversation. To encode non-local contextual information between each utterance, we apply max pool operation to the hidden states of each utterance-level BiLSTM (local LSTM), and then feed the sequence $c = \{c_1, c_2, ..., c_N\}$ to a conversation-level BiLSTM (global LSTM). The operation of global LSTM is the same as Eqs. (3.1) to (3.3)

### Graph Construction

#### Node Construction

In our model, we design a heterogeneous graph network containing five types of nodes: utterance nodes, type nodes, word nodes, speaker nodes, and argument nodes. Each type of node is used to encode a type of information in the dialogue.

In the task, only word nodes, speaker nodes and argument nodes areprobable to attend the final classification process. In other words, only these types of nodes are possible arguments. For simplicity, we name them as basic nodes in our illustration.

#### Utterance and Type Nodes

Utterance nodes are initialized by the utterance embeddings which we obtain from the utterance encoder. They are connected with the basic nodes which constitute the utterance. Type nodes represent the entity types of words in an utterance, where a variety of named and numeric entities, such as PERSON or LOCATION, are included. Since one mention may have different types in one conversation, type nodes can facilitate information integration. 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. Type nodes are connected with the basic nodes that possess the type attribute in the conversation. Each type node is initialized with a specific type of embedding. We believe that type attribute has a positive influence on the relation inference process.

### Basic Nodes

Word nodes represent the vocabulary of a conversation. Each word node is connected with a corresponding utterance and all the possible types that the word may have in the conversation. We initialize the states of word nodes with GloVe [26].

Speaker nodes represent each unique speaker in the conversation. Each speaker node is connected with the utterances by the speaker himself/herself. This type of node is initialized with some specific embeddings and can gather information from different speakers.

Argument nodes are two special nodes that are used to encode relative positional information of argument pairs. There are two argument nodes in each graph in total. One stands for the subject argument and the other represents the object argument. Similarly, argument nodes are also encoded by specific embeddings.

#### 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 edges. Each edge has a specific type. These edges are randomly initialized except for the utterance-word edge.

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

#### Graph Attention Mechanism

We use graph attention mechanism [25] to aggregate neighboring information to the target node. Suppose we have a node $i$ and some neighborhood nodes $j$, the graph attention mechanism can be described as:

$$ F(h_i, h_j) = \text{LeakyReLU}(a^T(W_i h_i, W_j h_j; E)) $$

$$ a_{ij} = \text{softmax}(F(h_i, h_j)) $$

$$ \exp(F(h_i, h_j)) $$

$$ \sum_k \exp(F(h_i, h_k)) $$

$$ h_i' = \sum_{k=1}^{K} \sigma(\sum_j a^k_{ij} W^k_j h_j) $$
where \( h_i \) and \( h_j \) are representations of node \( i \) and nodes \( j \), \( W_i, W_j, W_q \) and \( a' \) are trainable weight matrices, \( E_g \) is the edge weight matrix that is mapped to the multi-dimensional embedding space, \( a_j \) is the attention weight between \( i \) and \( j \), \( \sigma \) is an activation function, and \( \| \) is concatenation operation.

### Message Propagation

As shown in Fig. 2, there are five layers in our proposed graph module, where each layer represents an aggregation. There are four types of layers that we mark in the figure. Layer A and Layer D contain the message propagation between utterance nodes and basic nodes, and similarly, Layer B and Layer C are the message propagation between basic nodes and type nodes. We would call the whole message propagation path meta path. Different meta path strategies may lead to different performances.

Our meta path in this work can be described as follows: First, we use utterance nodes to update word nodes, speaker nodes, and argument nodes; secondly, the updated word nodes and argument nodes pass messages to type nodes; then type nodes conversely update the word nodes and argument nodes; secondly, the updated word nodes 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 type nodes.

Following [22], we add a residual connection [27] to avoid gradient vanishing during updating:

\[
\hat{h}_i = h_i + h'_i \tag{9}
\]

where \( h_i \) is the output learned in the graph attention layer, and \( h'_i \) is the original input of the graph attention layer.

In message passing, except for graph attention operation, there is also a two-layer feed-forward network which can be denoted as:

\[
h_i^{\text{new}} = \text{FFN}(\hat{h}_i) \tag{10}
\]

Suppose we have the initial embeddings of utterance nodes, basic nodes and type nodes, denoted as embedding matrices \( H_u = \{H_u, H_b, H_t\} \), the message propagating process can be written as:

\[
H_b^1 = \text{GAT}(H_u^0, H_b^0) \tag{11}
\]

\[
H_t^1 = \text{GAT}(H_u^0, H_t^0) \tag{12}
\]

\[
H_b^2 = \text{GAT}(H_b^1, H_t^1) \tag{13}
\]

\[
H_u^1 = \text{GAT}(H_u^0, H_b^2) \tag{14}
\]

\[
H_b^3 = \text{GAT}(H_b^2, H_u^1) \tag{15}
\]

where the GAT operation is the same as Eqs. (3.5) to (3.10) The superscripts represent the \( n^{th} \) update of the matrix and 0 marks the initial state.

### Relation Classifier

After the message propagation in the heterogeneous graph, we obtain new representations of all entities. We select the argument nodes \( \tau_x \) and \( \tau_y \), 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:

\[
e'_x = \text{maxpool}(\tau_x); \text{maxpool}(e_x) \tag{16}
\]

\[
e'_y = \text{maxpool}(\tau_y); \text{maxpool}(e_y) \tag{17}
\]

\[
e' = [e'_x; e'_y] \tag{18}
\]

\[
P(r|e'_x, e'_y) = \sigma(W_e e' + b_e)_r \tag{19}
\]

where \( P(r|e'_x, e'_y) \) is the probability of relation type \( r \) given argument pair \((e_x, e_y)\), \( W_e \) and \( b_e \) are linear transformation weight and bias vector, \text{maxpool} \) is max pooling operation, and \( \sigma \) is sigmoid function.

### Experiments

#### Dataset

We evaluate the proposed framework on the DialogRE dataset [10], which contains 1788 dialogues and 10,168 relational triples. The data statistics are shown in Table 1. DialogRE is adapted from the complete transcripts of Friends, a widely used corpus in dialogue research these years [28–31], 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 argument

#### Table 1 Statistics of the DialogRE dataset

|                | Train | Dev  | Test |
|----------------|-------|------|------|
| #Conversations  | 1073  | 358  | 357  |
| #Argument Pairs | 5963  | 1928 | 1858 |
| Average dialogue length | 229.5 | 224.1 | 214.2 |
| Average # of turns      | 13.1  | 13.1 | 12.4 |
| Average # of speakers   | 3.3   | 3.2  | 3.3  |
pair \((x, y)\). In the experiments, the dataset is partitioned into train, dev, and test sets with a roughly 60/20/20 ratio. Following the evaluation metrics of DialogRE, we report macro \(F_1\) scores of the proposed model and all the baselines in both the standard and conversational settings. In the following sections, we use \(F_1_c\) to represent \(F_1\) scores in the conversational setting.

Table 2 shows statistics of relation labels in DialogRE dataset. In the train set and test set, there are 35 types of relations, while in the dev set, there are 37 types. ‘gpe:birth_in_place’ and ‘per:place_of_birth’ only exist in the dev set.

### Table 2 Statistics of relation labels in DialogRE dataset

| Relation Type                  | Quantity       | Percentage (%) |
|--------------------------------|----------------|----------------|
|                               | train | dev | test | train | dev | test |
| per:alternate_names            | 1319  | 410 | 409  | 22.12 | 21.26 | 22.01 |
| unanswerable                   | 1308  | 404 | 388  | 21.94 | 20.95 | 20.88 |
| per:girl/boyfriend             | 502   | 170 | 136  | 8.42  | 8.82  | 7.32  |
| per:positive_impression        | 476   | 149 | 138  | 7.98  | 7.73  | 7.43  |
| per:friends                    | 444   | 156 | 122  | 7.45  | 8.09  | 6.57  |
| per:title                      | 250   | 86  | 78   | 4.19  | 4.46  | 4.20  |
| per:spouse                     | 204   | 72  | 54   | 3.42  | 3.73  | 2.91  |
| per:siblings                   | 196   | 64  | 58   | 3.29  | 3.32  | 3.12  |
| per:children                   | 171   | 55  | 48   | 2.87  | 2.85  | 2.58  |
| per:parents                    | 171   | 55  | 48   | 2.87  | 2.85  | 2.58  |
| per:negative_impression        | 156   | 46  | 56   | 2.62  | 2.39  | 3.01  |
| per:roomId                     | 140   | 44  | 24   | 2.35  | 2.28  | 1.29  |
| per:alumni                     | 110   | 38  | 34   | 1.84  | 1.97  | 1.83  |
| per:other_family                | 66    | 29  | 30   | 1.11  | 1.50  | 1.61  |
| per:works                      | 58    | 12  | 19   | 0.97  | 0.62  | 1.02  |
| per:age                        | 53    | 15  | 10   | 0.89  | 0.78  | 0.54  |
| per:client                     | 52    | 18  | 18   | 0.87  | 0.93  | 0.97  |
| per:place_of_residence         | 49    | 12  | 23   | 0.82  | 0.62  | 1.24  |
| gpe:residents_of_place         | 49    | 12  | 23   | 0.82  | 0.62  | 1.24  |
| per:boss                       | 49    | 13  | 12   | 0.82  | 0.67  | 0.65  |
| per:subordinate                 | 49    | 13  | 12   | 0.82  | 0.67  | 0.65  |
| per:visited_place              | 48    | 20  | 25   | 0.80  | 1.04  | 1.35  |
| gpe:visitors_of_place          | 48    | 20  | 25   | 0.80  | 1.04  | 1.35  |
| per:employee_or_member_of       | 46    | 11  | 15   | 0.77  | 0.57  | 0.81  |
| org:employees_or_members        | 46    | 11  | 15   | 0.77  | 0.57  | 0.81  |
| per:neighbor                   | 40    | 14  | 12   | 0.67  | 0.73  | 0.65  |
| per:place_of_work               | 37    | 9   | 25   | 0.62  | 0.47  | 1.35  |
| per:pet                        | 30    | 10  | 8    | 0.50  | 0.52  | 0.43  |
| per:acquaintance                | 26    | 12  | 34   | 0.44  | 0.62  | 1.83  |
| per:origin                     | 21    | 4   | 1    | 0.35  | 0.21  | 0.05  |
| per:dates                      | 20    | 14  | 6    | 0.34  | 0.73  | 0.33  |
| per:schools_attended           | 5     | 2   | 1    | 0.08  | 0.10  | 0.05  |
| org:students                   | 5     | 2   | 1    | 0.08  | 0.10  | 0.05  |
| per:major                      | 2     | 1   | 3    | 0.03  | 0.05  | 0.16  |
| per:date_of_birth               | 1     | 2   | 3    | 0.02  | 0.10  | 0.16  |
| gpe:birth_in_place             | 0     | 1   | 0    | 0     | 0.05  | 0     |
| per:place_of_birth              | 0     | 1   | 0    | 0     | 0.05  | 0     |
• Mutil-ATT-CNN [32] and ATT-BiLSTM [33] apply attention mechanism to CNN and BiLSTM.

• AGGCN [34] directly builds up a graph convolutional network whose structure is identical to the full dependency tree of each sentence. The network takes self-attention weights as soft edges and achieves state-of-the-art results in various relation extraction tasks.

• LSR [11] adopts a variant of Kirchhoff’s Matrix-Tree Theorem [35, 36] to induce the latent dependency structure of each document and then feeds the latent structure to a densely connected graph convolutional network for relation inference.

Settings and Hyper-parameters

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 3 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         | 10    |
| Learning rate                  | 0.0005|
| Batch size                     | 16    |
| Edge embedding dimension      | 50    |

| Model                          | #params | Dev (%) | Test (%) |
|--------------------------------|---------|---------|----------|
| Majority [10]                  |         | 38.9    | 35.8     |
| CNN [10]                       | 0.7M    | 46.1    | 48.0     |
| LSTM [10]                      | 2.1M    | 46.7    | 47.4     |
| BiLSTM [10]                    | 4.1M    | 48.1    | 48.6     |
| ATT-BiLSTM [33]                | 4.2M    | 48.2    | 49.1     |
| Multi-ATT-CNN [32]             | 0.7M    | 48.3    | 49.5     |
| LSR [11]                       | 20.5M   | 44.5    | 44.4     |
| AGGCN [34]                     | 3.7M    | 46.6    | 46.2     |
| DHGAT (Ours)                   | 4.0M    | 57.7    | 56.1     |

Simulating that logical connections are not locally compact within adjacent sentences, instead, they are spread over the whole conversation. Our proposed model 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 comparable to the BiLSTM models, which is only 4.0M.

Ablation Study

To understand the impact of our model’s components, we perform ablation studies using our proposed model on the DialogRE dataset. The ablation results are shown in Table 5. First, we remove local LSTM and global LSTM.

The dropping accuracy proves that the contextual encoder plays an important role in semantic feature extraction. Second, we remove the specific argument nodes and have observed that $F_1$ and $F_{1c}$ scores decrease to 55.0% and 50.2% on the test set. This proves that our design on argument nodes effectively synthesizes 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. The scores record a decrease under all these experiment settings. Notably, removing POS embedding leads to even about 2% drops in all the evaluation metrics.

Effect of the Meta Path

We test the performance of our message propagation strategy via changing meta path strategies. In our proposed model, there are five layers in the heterogeneous graph. Those basic
nodes, corresponding to different types of words, speakers, and arguments, are updated totally three times, i.e., they are first updated by utterance nodes, second updated by type nodes, and ultimately updated by utterance again. To investigate the meta path’s effect, we compare our proposed five-layer graph module with different strategies where the numbers of layers are one, seven, and nine in Table 6. In Strategy 1, we only set up one Layer A, where the basic nodes are updated by the initialized utterance nodes once. We observe that all the macro $F_1$ scores drop dramatically, showing the one-layer structure is not deep enough to grasp complex dependencies. To make node features more informative, we would add more layers. At this time, we are curious about how many layers the module should have to induct an optimal structure in this task. In Strategy 2 and Strategy 3, we design a seven-layer module and a nine-layer module, respectively. For Strategy 2, the order of layers is A-B-C-D-A-D-A, where A, B, C, and D are layer labels introduced in Fig. 2. Compared with our proposed module, scores on the validation set decrease by about 1%, and scores on the test set decrease 1.7% and 0.6% with the standard setting and the conversational setting, respectively. However, the module with nine layers in Strategy 3 shows a larger gap between itself and the best performance, where the order of layers is A-B-C-D-A-B-C-D-A. We think this is probably because the structure is so complicated, which causes an over-smooth problem and prevents itself from learning meaningful hidden representations.

| Model                     | Dev (%) | Test (%) |
|---------------------------|---------|----------|
|                           | F1      | F1c      | F1      | F1c      |
| Full model                | 57.7    | 52.7     | 56.1    | 50.7     |
| w/o Local BiLSTM          | 54.9    | 50.0     | 55.3    | 50.3     |
| w/o Global BiLSTM         | 54.7    | 50.2     | 53.5    | 48.7     |
| w/o Argument nodes        | 56.0    | 51.3     | 55.0    | 50.2     |
| w/o POS embedding         | 54.6    | 50.9     | 53.0    | 48.5     |
| w/o NER embedding         | 56.8    | 51.5     | 54.2    | 49.2     |
| w/o POS edge weights      | 56.9    | 52.4     | 54.7    | 50.4     |

### Case Studies

In the dataset, 95% of argument pairs span in at least two consecutive sentences instead of being restricted to the same sentence. Therefore, it is crucial that the model can tackle long-distance learning issues. Compared with the LSTM model, direct connections among different types of nodes in HGNN reduce the length of information propagation paths between pairs of argument nodes. Considering the following example in Fig. 3, subject a — ‘Mindy’ and object b — ‘Speaker 1’ share the relationship ‘per:friends’, which is 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 aggregates utterance-level information from its neighbor nodes containing a. the relation trigger ‘best friend’. b. in the BiLSTM model, the key information has to travel a long journey from the subject entity word to the object one as there are too many words between them in the context.

### Error Analysis

Type information involves in the information propagation process and thus affects the contents of output embeddings. The model is prone to make incorrect and biased predictions. If it fails to receive enough certainty from other information sources and then can only rely on the entity types of the two arguments. For example, given an argument pair of two human names, both are named entity type ‘PERSON’. Sometimes the model inclines to deem the relationship between the two arguments to be ‘per:alternate_name’ instead of the correct answer ‘per:alumni’ or ‘per:roommate’. This is because among all of these classes, ‘PERSON-PERSON’ is a preferable 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 the argument pair is not evident for judgment, entity bias misguides the model to the wrong classification results.

**Conclusion**

In this work, we present an attention-based heterogeneous graph network to deal with the dialogue relation extraction task in an inductive manner. This heterogeneous graph attention network has modeled multi-type features of the conversation, such as utterance, word, speaker, argument, and entity type information. On the benchmark DialogRE dataset, our proposed framework outperforms the strongest baselines and 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 work will focus on making use of latent relations between entities that exist in dialogue history to develop intelligent conversational agents.

**Funding** This research is supported by A*STAR under its RIE 2020 AME programmatic grant (Project no. RGAST2003, and Grantor reference no. A19E2b0098) and the Ministry of Education, Singapore, under its AcRF Tier-2 grant (Project no. T2MOE2008, and Grantor reference no. MOET2EP20220-0017). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of A*STAR and the Ministry of Education, Singapore.

**Data Availability** The curated used in this research is publicly available at https://github.com/nlpdata/dialogre.

**Declarations**

The authors did not receive support from any organization for the submitted work. This article does not contain any studies with human participants or animals performed by any of the authors. The authors declare no competing interests.

**References**

1. Peng N, Poon H, Quirk C, Toutanova K, Yih WT. Cross-sentence n-ary relation extraction with graph lstms. Trans Assoc Comput Linguist. 2017;5:101–15.

2. Quirk C, Poon H. 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: Long Papers. 2017;1:1171–82.

3. Li J, Sun Y, Johnson RJ, Sciaky D, Wei CH, Leaman R, et al. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. Database. 2016.

4. Wu Y, Luo R, Leung HC, Ting HF, Lam TW. Renet: A deep learning approach for extracting gene-disease associations from literature. In: International Conference on Research in Computational Molecular Biology. Springer; 2019;272–84.

5. Elsahar H, Vougiouklis P, Remaci A, Gravier C, Hare J, Laforest F, et al. 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.

6. Yao Y, Ye D, Li P, Han X, Lin Y, Liu Z, et al. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In:Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019;764–77.

7. Mesquita F, Cannavaccio M, Schmidek J, Mirza P, Barbosa D. 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); 2019;749–58.

8. Chia YK, Bing L, Poria S, Si L. RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction. In: Findings of the Association for Computational Linguistics: ACL 2022;45–57.

9. Wang D, Liu Y. A pilot study of opinion summarization in conversations. In: Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human language technologies. 2011;331–9.

10. Yu D, Sun K, Cardie C, Yu D. Dialogue-Based Relation Extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020;4927–40.

11. Nan G, Guo Z, Sekulic I, Lu W. Reasoning with latent structure refinement for document-level relation extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020;1546–57.

12. Sahu SK, Christopoulou F, Miwa M, Ananiadou S. Inter-sentence Relation Extraction with Document-level Graph Convolutional Neural Network. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019;4309–16.

13. Christopoulou F, Miwa M, Ananiadou S. 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). 2019;4927–38.

14. Pang B, Lee L. 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). 2004;271–8.

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

16. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In:5th International Conference on Learning Representations. 2017.

17. Zhang Y, Qi P, Manning CD. Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of the 2018 conference on empirical methods in Natural Language Processing. 2018;2205–15.

18. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Adv Neural Inf Process Syst 2017;5998–6008.

19. Cui D, Lam W. Graph transformer for graph-to-sequence learning. In: AAAI. 2020;7464–71.

20. Yao S, Wang T, Wan X. Heterogeneous Graph Transformer for Graph-to-Sequence Learning. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020;7145–54.

21. Linmei H, Yang T, Shi C, Ji H, Li X. 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). 2019;4823–32.

22. Wang D, Liu P, Zheng Y, Qiu X, Huang X. Heterogeneous Graph Neural Networks for Extractive Document Summarization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020;6209–19.

23. Chen W, Gu Y, Ren Z, He X, Xie H, Guo T, et al. Semi-supervised user profiling with heterogeneous graph attention networks. In: IJCAI. 2019;19:2116–22.

24. Peng H, Li J, Gong Q, Song Y, Ning Y, Lai K, et al. Fine-grained event categorization with heterogeneous graph convolutional networks. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. AAAI Press. 2019:3238–45.

25. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: 6th International Conference on Learning Representations. 2018.

26. Pennington J, Socher R, Manning CD. GloVe: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014;1532–43.

27. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016;770–8.

28. Chen HY, Zhou E, Choi JD. 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). Vancouver, Canada: Assoc Comput Linguist. 2017;216-25.

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

30. Yang Z, Choi JD. Friends QA: Open-domain question answering on TV show transcripts. In: Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue; 2019; 188–97.

31. Poria S, Hazarika D, Majumder N, Naik G, Cambria E, Mihalcea R. MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; 2019;527–36.

32. Wang L, Cao Z, De Melo G, Liu Z. Relation classification via multi-level attention cnns. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics Long Papers. 2016;1:1298–307.

33. Zhou P, Shi W, Tian J, Qi Z, Li B, Hao H, et al. Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th annual meeting of the association for computational linguistics (Short papers). 2016;2:207–12.

34. Guo Z, Zhang Y, Lu W. Attention Guided Graph Convolutional Networks for Relation Extraction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019;241–51.

35. Tutte WT. Graph Theory. In: Cren- don Press. 1984.

36. Koo T, Globerson A, Carreras X, Collins M. Structured prediction models via the matrix-tree theorem. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). 2007;141–50.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.