MFDG: a Multi-Factor Dialogue Graph Model for Dialogue Intent Classification

Jinhui Pang[1]✉, Huinan Xu[1], Shuangyong Song[2], Bo Zou[2], and Xiaodong He[2]

[1] Beijing Institute of Technology. Beijing 100081, China
{pangjinhui, xuhuinan}@bit.edu.cn
[2] JD AI Research. Beijing 100176, China
{songshuangyong,cdzoubo,hexiaodong}@jd.com

Abstract. Interest in speaker intent classification has been increasing in multi-turn dialogues, as the intention of a speaker is one of the components for dialogue understanding. While most existing methods perform speaker intent classification at utterance-level, the dialogue-level comprehension is ignored. To obtain a full understanding of dialogues, we propose a **M**ulti-**F**actor **D**ialogue **G**raph Model (MFDG) for Dialogue Core Intent (DCI) classification. The model gains an understanding of the entire dialogue by explicitly modeling multi factors that are essential for speaker-specific and contextual information extraction across the dialogue. The main module of MFDG is a heterogeneous graph encoder, where speakers, local discourses, and utterances are modelled in a graph interaction manner. Based on the framework of MFDG, we propose two variants, MFDG-EN and MFDG-EE, to fuse domain knowledge into the dialogue graph. We apply MFDG and its two variants to a real-world online customer service dialogue system on the e-commerce website, JD[3], in which the MFDG can help achieving an automatic intent-oriented classification of finished service dialogues, and the MFDG-EE can further promote dialogue comprehension with a well-designed knowledge graph. Experiments on this in-house JD dataset and a public DailyDialog dataset demonstrate that MFDG performs reasonably well in multi-turn dialogue classification.

Keywords: dialogue classification · core intent classification · graph neural network.

1 Introduction

There are increasing number of internet firms and platforms providing online customer services, thus creating lots of available multi-turn dialogues between customer service staffs and customers, which could be explored further for enhancing the user experience and satisfaction. Especially, the ability to recognize speakers' intentions, which is officially called Dialogue Intent (DI) classification [25,26], is essential to perceive the customers' requests across the dialogue. Most

[3] https://www.jd.com/

of works focus on the utterance-level DI classification, ignoring the dialogue-level comprehension. To promote the full understanding of dialogues, we bring forward a task, Dialogue Core Intent (DCI) classification, aiming to infer the core intention of the entire dialogue, such as refund promoting, product consultation, service complaint and etc. Early works regarded multi-turn dialogues as ordinary texts [1]. They simply concatenated the utterances in the dialogue, preventing them from learning the dialogue-level contextual dependency among utterances.

Dialogues have their own specific characteristics. As an example shown in Table 1, the speakers of a dialogue talk in a random order, breaking up the continuity of adjacent utterances in the dialogue. Moreover, topic transitions are common in human-human dialogues, which brings a new challenge of modeling the dependency among remote but interrelated utterances. The key point to address dialogue classification is adopting speaker-specific and contextual modeling [2].

Firstly, the speaker-specific dialogue modeling considers speaker information contained in the dialogue. It consists of two aspects: intra and inter speaker dependency. Intra-speaker dependency is used to reflect the affect that speakers have on themselves, which can contribute to the understanding of individual speakers. Inter-dependency implies the dynamic interactions among speakers. Modeling intra and inter speaker dependency in dialogues relies on plenty of factors, such as topic, speakers' personality and viewpoint [2]. Secondly, the contextual information coming from both neighbouring and distant utterances is indispensable for dialogue understanding. While the importance of neighbouring utterances is generally considered, it should be stressed that distant utterances can sometimes offer supplementary information when speakers refer to the same word that appears at former utterances.

In term of the above two points, DialogueGCN was proposed in [3], which built a directed graph to model both speaker dependency and contextual information in the dialogue. Later, other works inherited the graph modeling pattern and introduced discourse relations [4,5], position encoding [6] to the dialogue graph for enhancing the understanding of utterances in the dialogue.

It reminds us to acquire the comprehension of the dialogue based on a dialogue graph. Besides, considering the lack of additional factors' annotation, we focus on the very nature of multi-turn dialogues and build a multi factor dialogue graph. Not like prior methods using edges to inject speaker dependency, we explicitly define speaker nodes to represent the contextual information of speakers in the dialogue.

Moreover, we find the consecutive utterances spoken by the same speaker are generally highly correlated and supplementary to each other. A real example is shown in Table 1. The customer speaks U1, U2 and U3 continuously to explain the problem he (she) faces, thus it's helpful to integrate them to know the background information of the customer. We add local discourse nodes to aggregate such consecutive utterances for generating a dialogue representation later. The multi factor dialogue graph we build has three types of nodes, namely utterance nodes, speaker nodes and local discourse nodes. And the graph has five different

Table 1. An example dialog from JD dataset. The core intent label of this dialogue is 'refund urging'. Bold font denotes the pre-defined entities coming from a well-designed knowledge graph for JD dataset.

	Speaker	Utterance
U1	Customer	Hello? I can not reach the merchant.
U2	Customer	I bought some bread with a shelf life of a week, and it has been 4 days after I ordered.
U3	Customer	I haven't receive the bread, but it is probably expired at that time.
U4	Staff	We are sorry for our neglect. We will connect the merchant right now.
U5	Customer	I demand for return.
U6	Staff	You can apply for **Refund of unreceived goods** on the app.
U7	Customer	I have tried. This needs the permission of the merchant and I can not reach him.
U8	Staff	We can apply for **Order dispute** for you.
U9	Staff	Then you need to provide some evidence, after that we help you with the refund.
U10	Customer	all right.

types of edges, i.e., speaker edges, local edges, local-speaker edges, utterance-order edges and local-order edges. By applying a Graph Convolution Network (GCN) to this graph, we propagate contextual information among multi factors and obtain a multi-factor representation of the dialogue.

To sum up, we propose the Multi-Factor Dialogue Graph model (MFDG) by explicitly modeling the relations among speakers, utterances and local discourses in dialogues. We believe that the representation contains richer information relevant to core intent than other graph-based and text classification models. The results are shown in Section 5.

Furthermore, we discover there exist entities that contain domain-specific knowledge in online customer service dialogues. As shown in Table 1, pre-defined entities 'Refund of unreceived goods' and 'Order dispute' always appear with the refund demand of the customer. It will be helpful to take advantage of such domain knowledge. We explore two ways to fuse the fine-grained entity information into our original model MFDG, namely MFDG-Entity Node (MFDG-EN) and MFDG-Entity Embedding (MFDG-EE). On the basis of MFDG, MFDG-EN adds entity nodes to the dialogue graph and considers the inclusion relations among utterances and entities, MFDG-EE combine the token-level and entity-level representations and generate knowledge-aware initial representations for utterance nodes.

In summary, our main contributions are as follows:

- We propose a novel model, MFDG, to infer the core intention of a multi-turn dialogue by obtaining a full understanding of the entire dialogue.
- We build a heterogeneous dialogue graph to model the interactions among multi factors in the dialogue. Especially, we create local discourse nodes to aggregate consecutive utterances spoken by the same speaker and add speaker nodes to explicitly capture the speaker information.

– Additionally, we propose two variants of MFDG to explore an appropriate way to fuse domain knowledge into the dialogue graph.

2 Related Work

In this section, we firstly introduce current deep learning models for text classification, as dialogue core intent classification is a specific type of text classification. Considering the lack of dialogue-level classification models, we then introduce related works for utterance-level dialogue classification from the following two perspectives: recurrence-based models and graph-based models.

– **Text Classification** Deep learning models have achieved state-of-the-art results in many domains, including a wide variety of NLP applications. TextCNN [7] firstly migrated Convolutional Neural Networks (CNN) from computer vision to natural language processing. CNN makes use of convolution kernel to generate latent semantic features across the sentences, and performs much better than traditional feature-based text classification models. However, CNN does not take sequential information among sentences into consideration. As Recurrent Neural Network (RNN) is designed to recognize the sequential characteristics of data, it's a powerful model for text, string and sequential data classification [8]. Furthermore, an attention-based Long Short-Term Memory (LSTM) network [9] was proposed to dynamically integrate text information.
– **Recurrence-based Models** Utterances of the dialogue are inherently sequential, then [10] proposed RNN and LSTM models for utterance intent classification task. DialogueRNN [12] used two Gate Recurrent Units (GRU) to track individual speaker states and global context across the dialogue. COSMIC [13] shared a similar network with DialogueRNN and incorporated different elements of commonsense to learn interactions between speakers participating in a dialogue.
– **Graph-based Models** Many recent utterance-level dialogue classification models utilized graph-based neural networks to adopt speaker-specific and contextual modeling. DialogueGCN [3] firstly leveraged self and inter-speaker dependency of the speakers in a graph-based framework to model a conversational context, treating each dialogue as a graph where each utterance is connected to its surrounding utterances. Based on DialogueGCN, RGAT [16] added relational positional encodings that provide RGAT with sequential information implying in the dialogue. Besides, some other methods [4,5] draw support from pre-defined discourse relations between utterances. Lately, DAG [22] attempted to combine the advantages of recurrence-based models and graph-based models, which designs a directed acyclic graph to model the connections between nearby and distant utterances.

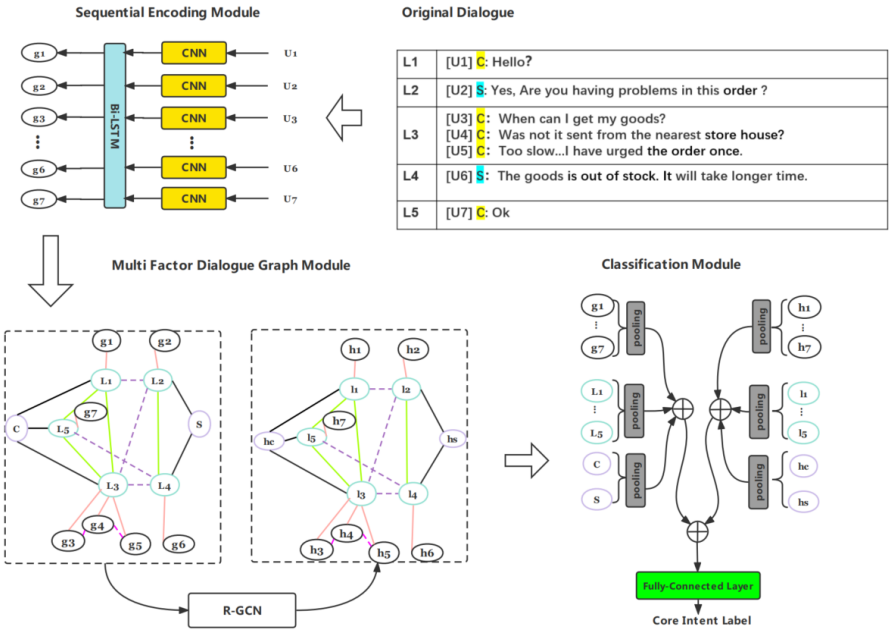

Fig. 1. The overall framework of MFDG. First, a sequential encoding module is used to obtain the initial representations of utterances in the dialogue. Then, in multi factor dialogue graph module we construct a dialogue graph consisting of three types of nodes and five types of edges. We utilize a graph convolutional network to update the nodes' features. Finally, three types of nodes are pooled and then concatenated together to be the dialogue representation which is fed to a fully connected layer for dialogue-level core intent classification.

3 Methodology

3.1 Problem Definition

In the following sections, let $D = \{U_1, U_2, ..., U_{N_u}\}$ be a dialogue with N_u utterances, and let there be N_s speakers $S = \{s_1, s_2, ..., s_{N_s}\}$ in dialogue D, where each utterance U_i is associated with the ID of its corresponding speaker by a mapping function $P(U_i)$. Given D, S and P, DCI attempts to predict the core intention label I of the dialogue.

3.2 Model

Now we present our Multi-Factor Dialogue Graph model (MFDG), which mainly consists of three modules as shown in Figure 1.

– **Sequential Encoding Module.** This module is used to produce context-dependent representations for utterances without considering speaker-specific

information, which will be used as the initial node features for the dialogue graph.
- **Multi Factor Dialogue Graph Module.** In this module, we organize the dialogue context as a heterogeneous graph. The detailed process of dialogue graph construction is below. Then a Relational Graph Convolutional Network (R-GCN) is applied to integrate contextual and speaker-specific information from multi factors in the dialogue graph.
- **Classification Module.** The last module predicts the core intention of a dialogue over the multi-factor involved dialogue representation.

Sequential Encoding Module Firstly, we follow [7] to make use of a CNN to extract features for each utterance. We use a simple CNN with one layer of convolution followed by max-pooling and a fully connected layer to learn the representations for the utterances.

Then, in order to obtain inherent contextual information among utterances, we feed the output of CNN to a Bidirectional Long Short-Term Memory (Bi-LSTM). Let $H = \{g_1, g_2, ...g_{N_u}\}$ be the output of the former CNN, the output features of utterances through Bi-LSTM can be represented as:

$$u_i = \left[\overleftarrow{LSTM(g_i)}; \overrightarrow{LSTM(g_i)}\right] \tag{1}$$

for $i = 1, 2, ...N_u$, where u_i is the sequential contextual feature for utterance U_i. Then $u_1, u_2, ..., u_{N_u}$ are used to initialize the node features of the dialogue graph.

Multi Factor Dialogue Graph Module In view of the characteristics of dialogues mentioned before, we explicitly model the interactions between utterances, speakers and local discourses. A heterogeneous graph with these three types of nodes is built to model the dialogue. Figure 2 is an example of dialogue graph for the original dialogue in Figure 1.

Same as prior works, each utterance in a dialogue is viewed as a node to represent the information of each turn in this dialogue, and the number of utterance nodes in a dialogue graph is same as that of turns in the dialogue. Then, speaker nodes are added for obtaining speaker information. The number of speaker nodes is decided by the speakers involved in the dialogue. In the online customer service scenario, there are usually two speakers, staff and customer. Besides, local discourse nodes denote the aggregated information for the sets of local longest continuous utterances uttered by the same speaker.

The initial representations of utterance nodes are the outputs of sequential encoding module. In addition, each speaker node initializes itself by averaging the representations of utterance nodes uttered by this speaker. Similarly, the mean of the representations of local longest continuous utterance nodes is used as the initial representation of the corresponding local discourse node. The number of speaker and local discourse nodes is denoted as N_s, N_l, respectively.

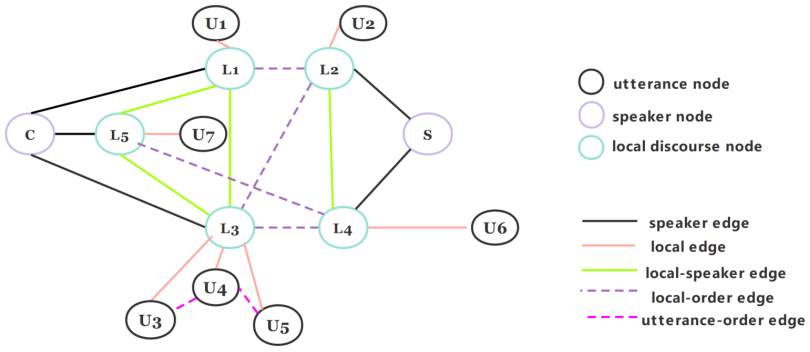

Fig. 2. Dialogue graph of the original dialogue in Figure 1. For brevity, we omit the self-loop edges. We set both utterance-level and local discourse-level context window to [1, 1].

In this heterogeneous graph, we define several different types of edges to indicate different aspects of knowledge. Here is the introduction of edges in the dialogue graph.

- **speaker edge**: Each of the speaker nodes is connected to all of its spoken local discourse nodes with the speaker edge so that the speaker node can learn speaker information in the dialogue.
- **local edge**: To strengthen the connections among local continuous utterances, we create the local discourse node for each of the sets of local longest continuous utterance nodes and connect the local discourse node with every utterance nodes in the set by the local edge.
- **local-speaker edge**: Despite using speaker edges to explicitly include the speaker information, local discourse nodes spoken by the same speaker are fully connected with the local-speaker edge to inject the intra-speaker dependency into the graph.
- **utterance-order edge** and **local-order edge**: To obtain the contextual information that comes from both neighbouring and distant utterances, two types of edges are created to introduce utterance-level and local discourse-level contextual information, respectively. Each utterance is connected to its contextual utterances by the utterance-order edge, and we set a utterance-level context window $[p, q]$ so that each utterance node has an edge with its past p utterances and latter q utterances. Besides, it should be emphasized that an utterance node only has utterance-order edges with utterance nodes which connect to the same discourse node with it. Likewise, each local discourse node is connected to its contextual local discourse nodes by local-order edges with a local discourse-level window size [m, n], which promotes the message passing among distant utterances.

Apart from above five types of edges, we also add self-loop edges for each node in the dialogue graph to facilitate effective computation. Therefore, there are totally six types of edges in the dialogue graph.

After acquiring the initial representation h_k for each node n_k and the edges among nodes, we feed the node features and the adjacent matrix into a graph neural network to obtain structural and semantic information of the dialogue. We apply R-GCN [15] to acquire the high-level hidden features with multi factors considered. The graph convolutional operation for node n_v at the $l+1$ layer can be defined as:

$$h_v^{(l+1)} = RELU\left(\sum_{r\in\mathcal{R}}\sum_{a\in N_r(v)} W_r^{(l)} h_a^{(l)} + b_r^{(l)}\right) \tag{2}$$

where $\mathcal{R}$ denotes different types of edges, $N_r(v)$ is the set of one-hop neighbors of node n_v under edge r, $W_r^{(l)}$ and $b_r^{(l)}$ denote the edge-specific learnable parameters at the l-th layer. Furthermore, $h_k^{(0)} = h_k$, for $k = 1, 2, ..., N$, where $N = N_u + N_s + N_l$ denotes the total number of nodes in the dialogue graph.

In addition, it is a natural thought that different types of edges can not be treated equally. We make use of the gating mechanism when aggregating information from different relations [14]. The simple idea is to compute a coefficient between 0 and 1 for each relation:

$$c_v^{(l)} = Sigmod(h_v^{(l)} W_r^{(l)}) \tag{3}$$

Therefore the message passing process for node n_v at the $l+1$ layer in the R-GCN can be overwrote as:

$$h_v^{(l+1)} = RELU\left(\sum_{r\in\mathcal{R}}\sum_{a\in N_r(v)} c_v^{(l)} W_r^{(l)} h_a^{(l)} + b_r^{(l)}\right) \tag{4}$$

Classification Module Finally, we concatenate the pooling results of output features of speaker nodes, utterance nodes and local discourse nodes at each GCN layer as hidden graph features. Here the pooling operation can be either max or mean pooling. Then, we concatenate the hidden graph features of all the GCN layers as the representation of the dialogue and makes the prediction using a fully-connected network.

3.3 Domain Knowledge Integration

Utterances of online customer service dialogues contain lots of domain-specific entities. An example is shown in Figure 1, where the entities come from a well-designed knowledge graph for JD dataset. Here we design two approaches to take advantage of the fine-grained entity information based on MFDG, MFDG-EN and MFDG-EE. Both of them utilize pre-trained knowledge graph embedding so we firstly give a brief introduction to knowledge graph embedding and then detail the two variant models.

Knowledge Graph Embedding Knowledge Graph (KG) is composed of triples in the form of *(head entity, relation, tail entity)*. Given all the triples in a KG, knowledge graph embedding aims to learn representation for each entity and relation that preserves structural information of the KG. There exist many translation-based knowledge graph embedding methods, such as TransE [18], TransH [19], TransR [20]. Considering those methods lack the ability of using the graph structures to enforce the local/global smoothness in the embedding spaces for entities and relations [21], we apply a simple R-GCN to acquire entity embedding from a pre-defined KG. Let us denote the pre-trained entity embedding as $[E_1, E_2, ..., E_j, ..., E_k]$, where K is the total number of entities of the KG and E_j is the generated embedding for entity e_j in the KG.

MFDG-EN The first variant of MFDG is proposed by adding entity nodes to the dialogue graph, named as MFDG-Entity Node (MFDG-EN). That is, every individual entity appearing in a dialogue is treated as a entity node in the dialogue graph, and each utterance node is connected to entity nodes that contained in the corresponding utterances by entity-utterance edges. Besides, the entity embedding generated from above is used to initialize the entity node. In this way, utterances containing the same entities can be indirectly connected by two consecutive entity-utterance edges, which was designed to promote message passing of domain knowledge in the dialogue graph.

MFDG-EE MFDG-Entity Embedding (MFDG-EE) leaves the dialogue graph unchanged, combining the token-level and entity-level representations and generating knowledge-aware initial representations for utterance nodes.

Here we use $U = t_{1:n} = [t_1, t_2, ..., t_n]$ to denote the raw sequence of an utterance in dialogue D, where n is the number of tokens in U. Then the token-level vectors for U can be obtained from a look-up word embedding table, which is denoted as $W = [w_1 w_2 ... w_n]$.

The entity-level vectors $E = [g_1 g_2 ... g_n]$ for U is generated as below:

$$g_i = \begin{cases} E_j, & \text{if } t_i \text{ is in the span of entity } e_j (j = 1, 2, ..., K) \\ \mathbf{0}, & \text{else} \end{cases}$$

Considering entity vectors are not in the same vector space with token vectors, we introduce a transformation function F for entity vectors:

$$F(E) = [F(g_1)F(g_2)...F(g_n)] \tag{5}$$

, where F can be either linear or non-linear mapping function.

Then we align and stack the token-level and entity-level embedding matrices as $M = [[w_1 F(g_1)][w_2 F(g_2)]...[w_n F(g_n)]]$. M will be fed into the sequential encoding module to compute knowledge-aware utterance representations. The rest of MFDG-EE is same as MFDG.

4 Experiment Setting

4.1 Datasets

We investigate several public dialogue datasets and find little information is available about dialogue-level labels. For this reason, we evaluate our MFDG model and its two variants on JD and DailyDialog datasets. The statistical information of them is shown in Table 2. Both the two datasets are composed of multi-turn dialogues where at least two speakers involve.

- **JD Dialogue dataset** This dataset is supplied by the customer service department of JD. Dialogues in this dataset are produced when customers consult the online customer service staffs about a series of issues. Each dialogue consists of several utterances with speaker annotations. The dialogues are annotated with one of 50 core intent labels, which are carefully designed by experts to summarize the essential intention of the customer during conversation. The dataset has 20,000 samples of dialogues, with a total of 437,060 utterances. We use 18,000 dialogues for training, 1,000 for validation, and the remaining for test.
- **DailyDialog** This dataset [23] reflects our daily communication way and covers various topics. Each dialogues in DailyDialog is annotated with one of the 10 certain topics, ranging from ordinary life to financial. It totally has 13,118 dialogues and 102,979 utterances. We use 11,118 dialogues for training, 1,000 for validation, and the remaining for test. Despite it does not have speaker annotations for utterances, we assume the utterances are spoken by two speakers one by one like previous works did.

Table 2. Statistical information of datasets. #Turn refers to the average number of utterances in a dialogue.

Dataset	#Dialogue	#Utterances	#Turn	#Class
JD dataset	20,000	473,060	23.65	50
DailyDialog	13,118	102,979	7.85	10

4.2 Evaluation Metrics

We adopt several widely used evaluation metrics, which are accuracy, H@3, H@5, macro-F1 and weighted-F1, to evaluate the performance of MFDG. Besides, we remove H@5 for DailyDialog, as there are just 10 classes in this dataset.

4.3 Baseline Methods

For the lack of dialogue-level classification model, we compare our model with several baseline methods for text classification, pre-trained models and some modified models of utterance-level classification models.

- **TextCNN** [7] This is a convolutional neural network based model for sentence classification. To acquire the features for dialogue-level classification, we add a max pooling layer to aggregate the utterances in the dialogue.
- **TextRNN** [8] In this method, a Bi-LSTM network is used to capture the contextual information from surrounding tokens in a text. We concatenate the utterances in a dialogue as an input of this model.
- **TextRNN-Att** [9] This model uses a Bi-LSTM with attention mechanism to automatically focus on the most informative words in a text. Likewise, we concatenate the utterances in a dialogue as an input of this model.

- **BERT-base**[4]**, Roberta-base**[5]**, ERNIE**[6] We use each of these three pre-trained models as an encoder for dialogues, following with a fully connected layer to acquire the dialogue-level labels.
- **Dialog-BERT** [27] Dialog-BERT designs three pre-training strategies to sufficiently capture dialogue exclusive features. We use the pre-trained model[7] as an encoder for dialogues, following with a fully connected layer to acquire the dialogue-level labels.

- **DialogueGCN** [3] DialogueGCN builds a graph for the dialogue where nodes represent individual utterances and the edges represent both the speaker and temporal dependency across the dialogue. DialogueGCN uses R-GCN as its graph encoder and initializes utterance features by using a CNN following a GRU. We modify DialogueGCN to a dialogue-level classification model by adding a max pooling layer to the graph neural network for acquiring representations of dialogues.
- **RGAT** [16] Based on the dialogue graph DialogueGCN builds, this module introduces position encodings to the graph to retain the sequential information contained in dialogues. RGAT uses the pre-trained BERT-base model to acquire the initial representations of utterance nodes. The modified operation is same as above.
- **DAG** [22] This model builds a directed acyclic graph for the dialogue with several carefully designed constraints on speaker dependency and positional relations. DAG introduces a directed acyclic graph neural network for utterance-level emotion recognition. Initial utterance embeddings in DAG is acquired form the pre-trained Bert-base model. The modified operation is same as above.

4.4 Other Settings

We choose cross entropy as the loss function for our model on two datasets. We take advantage of a cosine annealing schedule to dynamically modify the learning

[4] https://huggingface.co/bert-base-cased
[5] https://github.com/pytorch/fairseq/tree/main/examples/roberta
[6] https://github.com/nghuyong/ERNIE-Pytorch
[7] https://github.com/xyease/Dialog-PrLM

rate, and the initial learning rate is set to 1e-4. Adam optimizer is used in the training process with a batch size of 32 on both of the two datasets. JD dataset take the 300 dimensional Chinese Word Vectors [17] and DailyDialog use 300 dimensional pretrained 840B Glove vectors [24] as word embeddings. Then we set the CNN filter size to (3, 4, 5) with 50 out channels in each, following is a fully connected layer to get a 100 dimensional feature for each utterance. The hidden size of Bi-LSTM in the sequential encoding module is set to 100. We use 2-layer R-GCN to perform message passing on the dialogue graph. The utterance-level and local discourse-level window sizes are set to [5, 5] and [2,2], respectively. And We choose dropout rate that achieved the best score on each dataset by using validation data. Each training and testing process were conducted on a single Tesla P40 GPU. Every training process contain 60 epochs. The presented results are averages of 5 turns.

Besides, as for the knowledge graph resource that the two variant models MFDG-EN and MFDG-EE demand, we use a well-designed KG built by experts for JD Dialogue dataset. DailyDialog consists of daily communication dialogs and it's hard to design a KG for it, so we just extract general entities by *spaCy*[8] without pre-defined relations between entities and use the word embeddings of entities as the initial features of entity nodes.

5 Results and Analysis

5.1 MFDG comparing with Baseline methods

Table 3. Comparison with baseline methods on the JD Dialogue dataset; Bold font denotes the best performances.

Model	Acc	Top-3	Top-5	Macro-F1	Weighted-F1
TextRNN	49.10	74.90	84.00	38.62	46.52
TextRNN-Att	55.30	79.00	86.10	47.77	52.83
TextCNN	63.80	85.80	92.20	57.72	62.58
Bert-base	62.90	83.40	88.60	57.48	61.39
Robert-base	61.60	83.10	88.70	56.52	61.10
ERNIE	64.80	83.30	87.90	60.89	64.04
Dialog-BERT	63.70	87.70	93.40	55.09	61.21
DialogueGCN	61.30	83.90	90.80	52.48	58.70
RGAT	63.50	89.40	93.90	59.02	63.53
DAG	63.20	86.40	93.20	58.52	61.69
MFDG	66.50	89.30	94.40	60.64	65.30
MFDG-EN	65.00	88.60	94.00	60.71	64.00
MFDG-EE	**67.70**	**90.40**	**95.20**	**61.06**	**66.48**

We show the performance of MFDG and its variants with other baseline methods in Table 3 and Table 4. Our model outperforms text classification

[8] https://spacy.io/

baseline methods and other graph-based models. On the JD dataset, apart from MFDG-EE, MFDG achieves best Macro-F1 of 60.64%, Top-3 of 89.3%, Top-5 of 94.4%, and accuracy of 66.5%, which is 4.2% better than RGAT, and 2.9% better than the pre-trained model ERNIE. On the DailyDialog, MFDG achieves best Macro-F1 of 61.41% and Weighted-F1 of 72.22%.

It shows that graph-based models outperform most of the text classification models, as they adopt speaker-specific and contextual modeling for dialogue understanding, whereas text classification models treat the dialogue as an ordinary text without consider the characteristics of the dialogue. Besides, we notice that DialogueGCN perform worse than TextCNN, It demonstrates that DialogueGCN can obtain a good understanding of utterances, however mere modeling interactions between surrounding utterances leads to obvious losses of dialogue-level contextual information.

Besides, we notice MFDG underperforms Dialog-BERT on DailyDialog, otherwise outperforms Dialog-BERT on JD dataset. As the dialogues in JD dataset contain more utterances and speakers of dialogues in it talk in a random order, which is differ from dialogues in DailyDialog as the speakers talk one by one, we consider our MFDG shows its superiority in the real human-to-human multi-turn conversation scenarios.

With regard to the gap in performance between MFDG and other three graph-based models, it is important to understand the nature of these models. All of them build a dialogue graph and apply a GNN to train the model, whereas, other graph-based models only capture contextual information among utterances. MFDG adds other factors, speaker and local discourse, to the dialogue graph, modeling the contextual information of the dialogue form different levels, acquiring a more comprehensive understanding of the dialogue.

In addition, we notice that MFDG performs much better than other graph-based models on the real-world e-commerce dialogue dataset. As the dialogue in JD dataset contains more turns and is more complicated than that of DailyDialog, we believe our model MFDG contributes to enhancing the understanding of complex multi-turn dialogues in a real world scenario.

5.2 Ablation Study

We conduct ablation studies to evaluate the effectiveness of speaker nodes and local discourse nodes we add to the dialogue graph. The results are shown in Table 5.

Firstly, we remove speaker nodes and local discourse nodes from the dialogue graph in MFDG, leaving only the utterance-order edges accordingly. Without the two types of nodes, the performance of MFDG drops by 5.3% accuracy score on JD dataset and 3.44% accuracy score on DailyDialog. Besides, it should be mentioned that we find MFDG without considering speaker and local discourse nodes shares a close accuracy score with DialogueGCN, which can be rationally explained, as both of them model interactions between surrounding utterances.

Secondly, we remove the speaker nodes from the dialogue graph in MFDG, thus removing speaker edges accordingly. Without speaker nodes, the perfor-

Table 4. Comparison with baseline methods on DailyDialog.

Model	Acc	Top-3	Macro-F1	Weighted-F1
TextRNN	53.12	84.38	42.62	47.20
TextRNN-Att	68.20	93.40	50.17	66.36
TextCNN	71.60	93.30	55.29	69.39
Bert-base	70.20	93.00	59.00	69.01
Robert-base	72.90	95.20	60.67	71.54
ERNIE	71.90	93.30	53.12	70.55
Dialog-BERT	**74.00**	**94.90**	59.09	72.13
DialogueGCN	70.30	93.70	52.64	68.30
RGAT	72.30	93.40	55.32	70.31
DAG	72.30	92.90	56.18	70.58
MFDG	73.70	94.20	**61.41**	**72.22**
MFDG-EN	70.90	93.50	47.09	69.51
MFDG-EE	71.00	94.10	55.47	68.90

Table 5. Nodes ablation on two datasets. ✗ and ✓denotes nodes removed and added respectively.

speaker node	local discourse node	Acc(JD)	Acc(DailyDialog)
✗	✗	61.20(-5.3%)	70.26(-3.44%)
✗	✓	65.60(-0.9%)	71.60(-2.1%)
✓	✗	61.40(-5.1%)	73.60(-0.1%)
✓	✓	**66.50**	**73.70**

mance of MFDG drops by 0.9% accuracy score on JD dataset and 2.1% accuracy score on DailyDialog. This shows that speaker nodes help aggregating speaker-specific information in message passing of dialogues.

Lastly, we remove the local discourse nodes from the dialogue graph in MFDG, thus leaving only the utterance-order edges. In order to keep speaker nodes function in MFDG, we add utterance-speaker edges, which connect each speaker node with its corresponding spoken utterance nodes. Without local discourse nodes, the performance of MFDG drops by 5.1% accuracy score on the JD dataset and 0.1% accuracy score on the DailyDialog. The tiny drop on DialyDialog is because that speakers of the dialogue in DailyDialog talk one by one, forcing each local discourse node connect to only one utterance node, which can not show its superiority. And the drop on JD dataset shows that local discourse nodes are effective at aggregating multiple consecutive utterances spoken by the same speaker.

5.3 Variants of MFDG

As shown in Table 3 and Table 4, MFDG-EN obtains the accuracy score of 65.00% on JD dataset and 70.09% on DailyDialog, underperforming MFDG on the two datasets. It indicates that the addition of entity nodes leads to informa-

tion loss of the dialogue graph, as the features of entity nodes generated from KG are not in the same semantic space with other nodes in the graph.

For JD dataset, MFDG-EE outperforms MFDG on all the metrics, with a 1.2% promotion of accuracy score and 1.18% improvement of weighted-F1. The results prove the effectiveness of commonsense sense integration on dialogue classification. And it also shows the knowledge-aware representation method we design in MFDG-EE is an appropriate way to integrate entity information. However, MFDG-EN underperforms MFDG on DailyDialog. This is a predictable result as we use general entities for DailyDialog because of the lack of a well-designed KG.

6 Conclusion

In summary, we propose MFDG for dialogue core intent classification. MFDG is designed to obtain a full understanding of the dialogue by building a multi factor graph. Experimental results on two datasets demonstrate that MFDG outperforms other baseline methods. Furthermore, we propose MFDG-EE and MFDG-EN to fuse domain knowledge into the dialogue graph, the experiment results show that MFDG-EE can promote dialogue comprehension with a well-designed knowledge graph.

7 Acknowledgement

This work was supported by the National Key R&D Program of China under Grant No.2020AAA0108600 and Guizhou Province Science and Technology Plan Project-Research on Knowledge Management Technology Based on KG.

References

1. Ortega D, Vu N T. 2017. Neural-based Context Representation Learning for Dialog Act Classification. In Proc. of SIGDIAL, 2017.
2. Ghosal D, Majumder N, Poria S, et al. 2020. Utterance-level Dialogue Understanding: An Empirical Study. CoRR abs/2009.13902(2020).
3. Ghosal D, Majumder N, Poria S, et al. 2019. DialogueGCN: A Graph Convolutional Neural Network for Emotion Recognition in Conversation. In Proc. of EMNLP-IJCNLP, 2019. ACL, 154–164.
4. Feng X, Feng X, Qin B, et al. 2021. Dialogue Discourse-Aware Graph Model and Data Augmentation for Meeting Summarization. In Proc. of IJCAI 2021, 3808–3814.
5. Li J, Liu M, Zheng Z, et al. 2021. DADgraph: A Discourse-aware Dialogue Graph Neural Network for Multiparty Dialogue Machine Reading Comprehension. In Proc. of IJCNN, 2021. IEEE, 1–8.
6. Ishiwatari T, Yasuda Y, Miyazaki T, et al. 2020. Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In Proc. of EMNLP,2020. ACL, 7360–7370.

7. Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In Proc. of EMNLP 2014. ACL, 1746–1751.
8. Liu P, Qiu X, Huang X. 2016. Recurrent Neural Network for Text Classification with Multi-Task Learning. In Proc. of IJCAI, 2016. IJCAI/AAAI Press, 2873–2879.
9. Zhou P, Shi W, Tian J, et al. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In Proc. of ACL, 2016, Volume 2: Short Papers.
10. Suman V. Ravuri and Andreas Stolcke. 2015. Recurrent neural network and LSTM models for lexical utterance classification. In INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, 2015. ISCA, 135–139.
11. Qin L, Che W, Li Y, et al. 2020. Dcr-net: A deep co-interactive relation network for joint dialog act recognition and sentiment classification. In Proceedings of the AAAI Conference on Artificial Intelligence. AAAI Press, 8665-8672.
12. Majumder N, Poria S, Hazarika D, et al. 2019. DialogueRNN: An Attentive RNN for Emotion Detection in Conversations. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 6818–6825.
13. Ghosal D, Majumder N, Gelbukh A, et al. 2020. COSMIC: COmmonSense knowledge for eMotion identification in conversations. In Proc. of EMNLP 2020. ACL, 2470–2481, Online.
14. Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In Proc. of EMNLP, pages 1506–1515.
15. Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling relational data with graph convolutional networks. 2018. In Proc. of ESWC. Springer, 2018: 593-607.
16. Ishiwatari T, Yasuda Y, Miyazaki T, et al. Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations.2020. In Proc. of EMNLP. 2020: 7360-7370.
17. Li S, Zhao Z, Hu R, et al. 2018. Analogical reasoning on chinese morphological and semantic relations. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 138–143.
18. Bordes A, Usunier N, Garcia-Duran A, et al. 2013. Translating embeddings for modeling multi-relational data. In Advances in Neural Information Processing Systems. 2787–2795.
19. Wang Z, Zhang J, Feng J, et al. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In AAAI. 1112–1119.
20. Lin Y, Liu Z, Sun M, et al. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In AAAI, 2181–2187.
21. Yu D, Yang Y, Zhang R, et al. Knowledge embedding based graph convolutional network. 2021. In Proceedings of the Web Conference 2021. 2021: 1619-1628.
22. Shen W, Wu S, Yang Y, et al. Quan, Directed acyclic graph network for conversational emotion recognition. 2021. In Proc. of ACL/IJCNLP, 1551–1560.
23. Li Y, Su H, Shen X, et al. 2017. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In Proc. of IJCNLP, 986–995.
24. Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation. 2014. In Proc. of EMNLP. 2014: 1532-1543.
25. Guo D, Tur G, Yih W, et al. Joint semantic utterance classification and slot filling with recursive neural networks. 2014. IEEE Spoken Language Technology Workshop (SLT). IEEE, 2014: 554-559.
26. Ravuri S, Stoicke A. A comparative study of neural network models for lexical intent classification. 2015. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, 2015: 368-374.

27. Xu Y, Zhao H. Dialogue-oriented Pre-training. Findings of the Association for Computational Linguistics, Online Event, August 1-6, 2021.