Graph-Based Knowledge Integration for Question Answering over Dialogue

Jian Liu[1,2,3]*, Dianbo Sui[1,2]*, Kang Liu[1,2], Jun Zhao[1,2]
[1] National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing, 100190, China
[2] University of Chinese Academy of Sciences
[3] Beijing Jiaotong University, 100044, China
jianliu@bjtu.edu.cn; {dianbo.sui, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Question answering over dialogue, a specialized machine reading comprehension task, aims to comprehend a dialogue and to answer specific questions. Despite many advances, existing approaches for this task did not consider dialogue structure and background knowledge (e.g., relationships between speakers). In this paper, we introduce a new approach for the task, featured by its novelty in structuring dialogue and integrating background knowledge for reasoning. Specifically, different from previous "structure-less" approaches, our method organizes a dialogue as a "relational graph", using edges to represent relationships between entities. To encode this relational graph, we devise a relational graph convolutional network (R-GCN), which can traverse the graph's topological structure and effectively encode multi-relational knowledge for reasoning. The extensive experiments have justified the effectiveness of our approach over competitive baselines. Moreover, a deeper analysis shows that our model is better at tackling complex questions requiring relational reasoning and defending adversarial attacks with distracting sentences.

1 Introduction

Humans obtain information by engaging in conversations. Question answering (QA) over dialogue, a specialized machine reading comprehension (MRC) task (Hermann et al., 2015), aims to test the ability of a QA system to comprehend a dialogue, by asking it to answer questions about the dialogue. Consider the example shown in Table 1. Given a dialogue and a related question Q1: "*What is Joey going to do with Kathy tonight?*", the task requires a system to give the correct answer "*having a late dinner*".

U1	Chandler:	Hey-Hey-Hey! Who was that?
U2	Joey:	That would be Casey. We're going out tonight.
U3	Chandler:	Goin' out, huh? Wow! Wow! So things didn't work out with Kathy, huh? Bummer.
U4	Joey:	No. Things are fine with Kathy. [I'm having a late dinner with her tonight], right after my early dinner with Casey.

Q1	What is Joey going to do with Kathy tonight?	A1	having a late dinner
Q2	When will Joey have dinner with Casey?	A2	tonight

Table 1: Up: a dialogue from FriendsQA corpus (Yang and Choi, 2019). Down: two related questions with their answers. An evidence sentence for inferring A1 is given in [].

Compared with other MRC tasks, QA over dialogue is more challenging (Yang and Choi, 2019) owing to that conversations often involve complex relationships and background knowledge. In detail, studies show that a dialogue with 12 turns contains 6.1 co-reference chains (Zhou and Choi, 2018) and expresses 4.5 relationships (Yu et al., 2020) on the average. Therefore, to excel in this task, a QA system must master background knowledge for reasoning. Let us consider the reasoning process of Q1 in the above

* Equal Contribution

Proceedings of the 28th International Conference on Computational Linguistics, pages 2425–2435
Barcelona, Spain (Online), December 8-13, 2020

example. To find the correct answer, a QA system should not only locate the evidence sentence "*I'am having a late dinner with her*" (in U4) but also master *co-reference knowledge* that "*I*" refers to Joey (the speaker) and "*her*" refers to Kathy. While, how to effectively integrate the background knowledge in this task remains an open question. Existing approaches for this task (Yang and Choi, 2019; Li and Choi, 2020) did not consider background knowledge and only learned reasoning patterns from plain texts. This may expose them at risk of achieving sub-optimal results and becoming vulnerable facing adversarial attacks (i.e., models only learn shallow patterns for reasoning is fragile facing adversarial examples by adding distracting sentences (Jia and Liang, 2017)).

In this paper, we propose a new approach for QA over dialogue, featured by its novelty in structuring dialogue and integrating background knowledge — specifically, co-reference and relation knowledge — for reasoning. Different from previous "structure-less" methods, our approach structures the dialogue as a "relational graph", where nodes correspond to words in contexts and edges designate their relationships. The graph uses different types of edges to indicate different types of relations and thus is a heterogeneous graph. To encode this graph, we devise a model based on relational graph convolutional networks (R-GCN) (Schlichtkrull et al., 2018), which learns reasoning patterns considering the topology of the graph. We show in this way, background knowledge is effectively incorporated to guide the reasoning process of question answering.

To confirm the effectiveness of our method, we have conducted extensive experiments on a benchmark dataset FriendsQA (Yang and Choi, 2019). Experimental results demonstrate that our approach achieves superior performance over competitive baselines. Moreover, a deeper analysis reveals that, by integrating background knowledge, our approach is better than baselines at 1) tackling complex questions requiring relational reasoning, and 2) defending adversarial attack with distracting sentences. We have released our code at `https://github.com/jianliu-ml/dialogMRC` to encourage more studies in this research line.

To sum up, we make the following contributions:

- We propose a new approach for QA over dialogue, featured by its novelty in structuring dialogue and integrating background knowledge for reasoning.
- We consider both co-reference and relation knowledge for the task. And a R-GCN is devised to explore multi-relation data characteristics of a heterogeneous relational graph representing a dialogue. To our best knowledge, this is the first work introducing R-GCN to QA over dialogue.
- We set up a new state-of-the-art performance on the benchmark dataset. Moreover, results of robustness testing suggest that our method is robust against adversarial examples.

2 Related Work

QA over Dialogue. QA over dialogue is a specified MRC task (Hermann et al., 2015), which requires a system to answer questions regarding a dialogue. Many recent studies have benchmarked and advanced this task. To name a few, Reddy et al. (2019) introduce CoQA corpus, which measures MRC over one-to-one conversation. Ma et al. (2018) introduce a corpus based on transcripts of a TV show *friends* and focus on questions whose answers are PERSON entities. Sun et al. (2019) propose DREAM, which focuses on multiple-choice question answering over multi-turn dialogues. Yang and Choi (2019) extend the work of Ma et al. (2018) and propose FriendsQA, a dataset annotated open-domain questions and answers. QA over dialogue is recognized more challenging than general MRC tasks. In our study, we chose FriendsQA as the testbed, considering its diversity in different types of questions. Moreover, the extractive QA style is more suitable than the multiple-choice style for building practical QA applications. On this benchmark, the best reported method (Li and Choi, 2020) combines a pre-trained language model (Devlin et al., 2019) with an utterance-level pre-training strategy.

Knowledge Incorporation for MRC. Integrating background knowledge to enhance machine reading is a longstanding goal of artificial intelligence. In the task of MRC, previous studies (Yang and Mitchell, 2017; Mihaylov and Frank, 2018; Weissenborn, 2017; Bauer et al., 2018; Qiu et al., 2019) have exploited

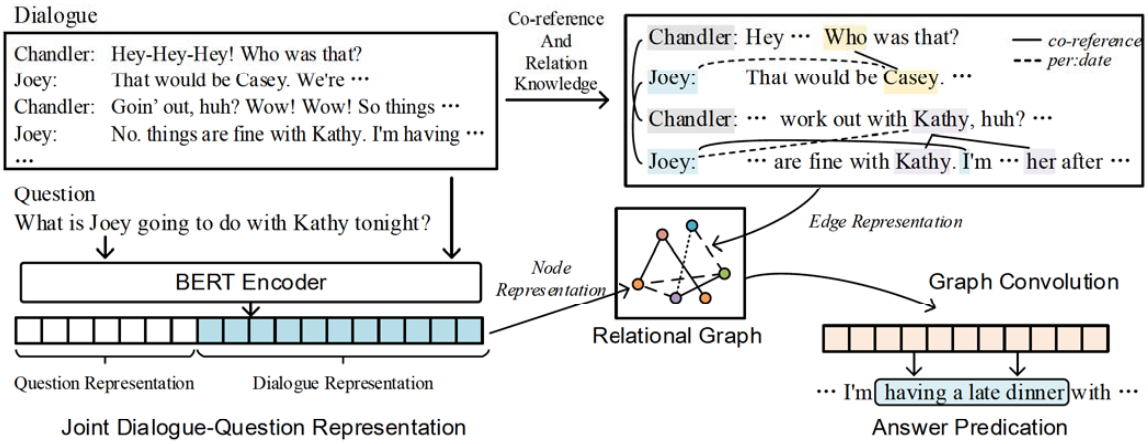

Figure 1: The overview of our approach, which structures the dialogue as a relational graph and integrates co-reference and relation knowledge for reasoning.

external knowledge. While, such work may not be applied to QA over dialogue, whose contexts are dynamic. It also worth noting that Qiu et al. (2019) adopt graph structure to model external knowledge, but in their work, the relation is not discerned, with a general "related_to" relation. By contrast, our approach uses a heterogeneous graph to incorporate different types of knowledge.

Graph Representation Learning. Graph neural networks (GNNs) (Kipf and Welling, 2016; Veličković et al., 2017; Schlichtkrull et al., 2017) provide an effective way to model graph-structure data and show promising results in many NLP problems (Vashishth et al., 2019; Gui et al., 2019; Liu et al., 2019; Qiu et al., 2019). Among all GNNs, Relational Graph Convolution Networks (R-GCNs) (Schlichtkrull et al., 2017) are variations of Graph Convolution Networks (GCNs) that are designed for modeling multi-relation data. To our knowledge, this is the first work introducing R-GCNs to model co-reference and relation knowledge for the task of QA over dialogue.

3 Approach

Figure 1 schematically visualizes our approach, which involves three major steps:

- Joint dialogue-question representation. In this step, the dialogue and question are jointly encoded to build their representations, and the dialogue representations are taken as the initial node representations of the relational graph.

- Graph-based knowledge integration, where the dialogue is organized as a relational graph and a R-GCN is proposed to integrate co-reference and relational knowledge for reasoning the answer.

- Answer span prediction. This module reasons over the knowledge enhanced representation and generates a text span as the answer to the question.

In the following illustrations, let $D = \{U_1, U_2, ..., U_N\}$ be a dialogue with N utterances. Each utterance U_i is associated with a speaker s_i. The texts in U_i can be denoted as $\{w_{i1}, w_{i2}, ..., w_{im}\}$ where w_{ij} is the jth token in U_i, and m is the length of U_i. Let a question be $Q = \{q_1, q_2, \ldots, q_L\}$ where L is the length of Q. Given D and Q, QA over dialogue requires to predict an answer a. Note a is restricted to be a (continuous) span in D (Yang and Choi, 2019).

3.1 Joint Dialog and Question Representation

We first encode D and Q into continuous representations to learn their joint representations. We adopt the BERT based QA architecture (Devlin et al., 2019) considering its effectiveness. Specifically, given

D and Q, we first construct an input sequence to concatenate them:

$$[\text{CLS}], \underbrace{q_1, q_2, \ldots, q_L}_{Q}, [\text{SEP}], \underbrace{[\text{S}], s_1, [/\text{S}], \overbrace{w_{11}, w_{12}, \ldots}^{U1}, [\text{S}], s_2, [/\text{S}], \overbrace{w_{21}, w_{22}, \ldots}^{U2}, ..., }_{D} [\text{SEP}] \quad (1)$$

where [CLS] and [SEP] are special tokens used in BERT; [S] and [/S] are special tokens used by our approach to indicate speakers' positions. We then split the above sequence into sub-word pieces according to Byte-Pair Encoding (BPE) (Sennrich et al., 2016) and adopt BERT to encode the sequence[1]. We take the last hidden layer of BERT as the joint representation of D and Q, denoted as $\boldsymbol{H} \in \mathcal{R}^{T \times d}$, where T is the length of the extended input sequence (regarding sub-word pieces), and d is the hidden dimension of BERT. $\boldsymbol{H}$ can be divided as $\boldsymbol{H}_D$ and $\boldsymbol{H}_Q$ to indicate dialogue-specific and question-specific representations. $\boldsymbol{H}_D$ is used to initialize the node representations in the relational graph.

3.2 Graph-Based Knowledge Integration

Graph-based knowledge integration involves relational graph construction, knowledge integration via graph convolution, and representation fusion.

Relational Graph Construction. We first organize dialogue contexts as a "relational graph", where the nodes correspond to words in D, and the edges reflect their relationships. We consider two types of relationships: 1) co-reference knowledge (Chen et al., 2017), which designates expressions referring to the same entity, and 2) relation knowledge (Yu et al., 2020), which reflects semantic relations between two entities (We refer to § 4.1 for how we obtain such knowledge). A heterogeneous graph is proposed to model the knowledge, which uses different types of edges to indicate different types of knowledge. We also add self-loop edges in the graph to facilitate effective computation (Schlichtkrull et al., 2017). Thus, the total number of different types of edges is 1 + 1 + N_r (self-loop + co-reference + number of semantic relation).

Knowledge Integration via Graph Convolution. We next encode the relational graph via a relational graph convolution network (R-GCN), to allow knowledge integration. In R-GCN, the representation of a node is computed by gathering information from its neighbor nodes, using the following rules:

$$\boldsymbol{h}_i^{(l+1)} = \sigma\Big(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}^r(i)} \frac{1}{c_{i,r}} \boldsymbol{W}_r^{(l)} \boldsymbol{h}_j^{(l)}\Big) \quad (2)$$

where $\mathcal{N}^r(i)$ is the neighbor set of node i regarding a relation r; $\boldsymbol{h}_j^{(l)}$ is the representation of note j at the lth layer; $\boldsymbol{W}_r^{(l)}$ corresponds to the parameter matrix associated with a relation r at the lth layer. $c_{i,r}$ is the normalization term that equals to the size of $\mathcal{N}^r(i)$. σ is the sigmoid function. In this way, the representation of a node[2] is encoded by considering all nodes that have relationships with it, and meanwhile different types of relations are considered. We use $\boldsymbol{H}_D$ as the initialized representation of each node in the relational graph. And the overall graph would be updated k times (where k is a hyper-parameter tuned on the development set) to allow long-range dependency. The obtained representations are denoted by $\boldsymbol{H}_G$.

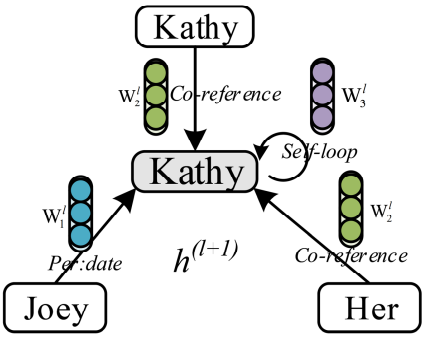

Figure 2: Illustration of convolution computation in the relational graph. The shaded node indicates the currently focused node, and the others are nodes having relations with it.

Representation Fusion. In practice, we found combing $\boldsymbol{H}_G$ with the original representation $\boldsymbol{H}_D$ yields better performance. Perhaps because that $\boldsymbol{H}_G$ may not preserve the position information in the dialogue contexts well. The final fused representation is computed as:

$$\boldsymbol{H}_{enh} = \alpha \boldsymbol{H}_G \oplus (1 - \alpha) \boldsymbol{H}_D \quad (3)$$

[1]The segmentation and position embedding are also added, following standard BERT.

[2]In cases where a word is cut into many sub-word pieces, only the leading sub-word is considered as the node in graph.

where $\oplus$ is the element-wise add computation and α is a hyper-parameter tuned on the development set[3]. Finally, $\boldsymbol{H}_{enh}$ is used as the final dialogue representation to infer the answer.

3.3 Answer Span Prediction

To generate the answer, we compute two probability vectors containing the starting and ending positions of answer by taking $\boldsymbol{H}_{enh}$ as the input:

$$\boldsymbol{o}_{start} = \text{softmax}(\boldsymbol{H}_{enh}\boldsymbol{w}_{start}); \qquad \boldsymbol{o}_{end} = \text{softmax}(\boldsymbol{H}_{enh}\boldsymbol{w}_{end}); \tag{4}$$

where $\boldsymbol{w}_{start}$, $\boldsymbol{w}_{end}$ are learnable parameters. We rank all *legal* spans (i.e., the starting position should be ahead of ending position) based on their summed starting and ending probabilities and select the one with the highest value as the answer span. We map BPE positions to original positions to generate the answer.

3.4 Training and Optimization Strategy

We adopt cross-entropy loss to train our model. Specifically, the training loss function regarding a specific (dialogue, question, answer) triple (D, Q, a) is:

$$\mathcal{L}(D, Q, a) = -(\log(\boldsymbol{o}_{\text{start}}[a_s]) + log(\boldsymbol{o}_{\text{end}}[a_e])) \tag{5}$$

where a_s and a_e indicate the starting and ending positions of the ground-truth answer a. The overall training loss sums up cross-entropy loss of each training instance in the training set. We adopt Adam (Kingma and Ba, 2014) to optimize our model and a linear decaying strategy to smooth the training.

4 Experimental Setups

4.1 Datasets and Evaluations

Our experiments are conducted on a benchmark dataset FriendsQA (Yang and Choi, 2019), which annotates QA pairs on transcripts of a TV show friends. We split the dataset as training/developing/test set following the setting of Li and Choi (2020). The co-reference knowledge is obtained by aligning FriendsQA with Character Identification project (Zhou and Choi, 2018); the relation knowledge is obtained by aligning FriendsQA with DialogRE (Li and Choi, 2020), which defines 36 different semantic relations (e.g., per:father, per:date) between entities. The data statistics are shown in Table 2.

	Episodes	# Doc.	# Question	# Answer	# Coref	# Relation
Training Set	1-20	973	9,791	16,352	41116	8214
Developing Set	21-22	113	1,189	2,065	4200	690
Test Set	23-*	136	1,172	1,920	5476	842

Table 2: Data statistics of the FriendsQA benchmark. # Coref and # Relation denote the number of co-reference and relation knowledge.

For evaluation, we adopt three evaluation metrics: utterance matching (UM), which evaluates whether the predicted answer matching the utterance, span matching (SM), which treats an answer as a bag-of-token, and conducts a set-level token matching, and exact matching (EM), which measures whether a prediction matches the ground-truth answer exactly. We also adopt three training strategies, shortest-answer strategy, longest-answer strategy, and multiple-answer strategy, following Yang and Choi (2019; Li and Choi (2020), to evaluate our method.

4.2 Implementation Details

In our implementation, we choose BERT base architecture, with 12 layers and 768 hidden units, same as previous methods to ensure comparability (Yang and Choi, 2019; Li and Choi, 2020). The hidden

[3]We have also tried a more adaptive strategy, i.e., learning the weight via a gate mechanism, but with no improvement.

Model	SA Strategy			LA Strategy			MA Strategy		
	UM	SM	EM	UM	SM	EM	UM	SM	EM
R-NET	44.3	35.1	22.4	46.4	38.2	22.9	45.3	36.6	24.0
R-NET + Graph (ours)	46.2	44.1	24.4	51.2	42.3	26.7	55.3	44.2	28.7
BERT	68.5	57.2	41.6	69.1	58.3	42.2	70.2	59.3	43.3
BERT + Graph (ours)	72.8	63.6	44.8	72.7	61.9	44.6	73.4	64.3	46.4
BERT_{pre}	69.0	61.0	45.2	68.8	61.4	45.2	71.3	61.2	45.6
BERT_{pre} + Graph (ours)	**73.3**	**65.2**	**47.5**	**72.3**	**65.0**	**47.1**	**74.1**	**65.5**	**48.1**
SoTA (2020)	-	-	-	-	-	-	73.3†	63.1†	46.8†

Table 3: Results on the test set of FriendsQA. SA Strategy, LA Strategy, and MA Strategy are three training strategies using the shortest answer, the longest answer, and all of the answers for training. The best results are denoted in bold. † denotes that the results are directly taken from the original paper.

dimension of R-GCN is set as 60, chosen from 50 to 100. The layers of R-GCN, k, is set as 3, chosen from 1 to 5. The learning rate is set as 1.0×10^{-5}. The balance factor is set as 0.5, chosen from 0.1 to 0.9. We use Deep Graph Library (DGL)[4] to build the graph and implement graph model.

4.3 Baseline Models

We compare our model with the following baseline models: BERT, the standard BERT MRC model. BERT_{pre} is a model that uses dialog contexts to pre-train BERT (Li and Choi, 2020), which corresponds to the best reported method (denoted as SoTA). We also compare with R-Net (Wang et al., 2017), the earlier SoTA model achieving the 1st place on the SQuAD leaderboard, which builds representations for questions and evidence passages via a self-matching mechanism. Our model is denoted as + Graph (e.g., BERT_{pre} + Graph indicates using BERT_{pre} as basic encoding model).

5 Experimental Results

The results of comparing our approach with baselines models are shown in Table 3. Here we adopt golden co-reference and relation knowledge to build the relational graph (The results of using system predicted results are shown in § 7.1). From the results, our approach consistently outperforms models without leveraging background knowledge, regarding each evaluation metric. Moreover, our approach also outperforms the previous best-reported system (Li and Choi, 2020), especially in SM evaluation, setting up a new state-of-the-art. Among different training strategies, the multi-answer strategy yields better results, as expected, as it can leverage more data for training compared with other training strategies. It is also worth noting that pre-training on the dialogue datasets can improve the performance.

6 Discussion

We further investigate the performance of our model on different types of questions and conduct a robustness testing to understand why our approach is effective.

6.1 Results on Different Question Types

We compare BERT_{pre} and our model BERT_{pre}+G on different types of questions. The results are shown in Table 4. From the results, among all factoid questions (*Who*, *Where*, *When*, and *What*), our approach is especially excelled in answering *Who* and *What* questions, and outperforms BERT_{pre} by considerable margin. The reason may be that *Who* and *Why* questions are more relation-related, which are difficult for BERT_{pre} reasoning over plain texts. Moreover, our method demonstrates very promising results in answering *Why* questions (+6.5% over BERT_{pre}). This implies that answering *Why* questions should considering background knowledge, where structure-less methods such as BERT_{pre} are difficult to master such knowledge.

		Who (18.82%)	Where (18.16%)	When (13.57%)	What (18.48%)	How (15.32%)	Why (15.65%)
UM	BERT_{pre}	76.2	81.0	70.9	71.1	56.1	66.4
	BERT_{pre}+G	79.5 (↑2.7)	83.8 (↑1.2)	73.6 (↑2.7)	73.3 (↑2.2)	56.6 (↑0.5)	72.9 (↑**6.5**)
SM	BERT_{pre}	60.1	75.1	61.7	65.9	50.9	49.6
	BERT_{pre}+G	66.9 (↑**6.8**)	77.2 (↑2.1)	64.8 (↑3.1)	70.9 (↑**5.0**)	52.2 (↑1.3)	57.2 (↑**7.6**)
EM	BERT_{pre}	53.5	64.3	44.3	49.0	31.3	25.6
	BERT_{pre}+G	56.7 (↑3.2)	66.5 (↑2.2)	43.6 (↓0.7)	52.4 (↑3.4)	28.9 (↓2.4)	33.8 (↑**8.8**)

Table 4: Results on UM, SM, and EM on different types of questions (with percentages shown within parentheses) in FriendsQA. Large improvements (i.e, over 5.0) are denoted in **bold**.

		Overall			What Question			Who Question		
Model	Sett.	UM	SM	EM	UM	SM	EM	UM	SM	EM
BERT_{pre}	ORG	71.3	61.2	45.6	71.1	65.9	49.0	76.2	60.1	53.5
	ATT	69.2	56.3	41.1	66.4	61.0	44.2	70.9	55.7	48.5
	Δ	(↓2.1)	(↓**4.9**)	(↓**4.5**)	(↓**4.7**)	(↓**4.9**)	(↓**4.8**)	(↓**5.3**)	(↓4.4)	(↓**5.0**)
BERT_{pre}+G	ORG	74.1	65.5	48.1	73.3	70.9	52.4	79.5	66.9	56.7
	ATT	72.8	63.6	44.8	71.0	68.5	51.2	74.3	63.5	52.9
	Δ	(↓1.3)	(↓1.9)	(↓3.3)	(↓2.3)	(↓2.4)	(↓1.2)	(↓**5.2**)	(↓3.4)	(↓3.8)

Table 5: Robustness probing on the test set of FriendsQA. In the column of setting (Sett.), ORG indicates results on the original test set; ATT indicates results on the adversarial attacked test set; Δ indicates the performance gap. Performance gap larger than 4.5 is shown in **bold**.

6.2 Results of Robustness Testing

We conduct a robustness testing on our model and BERT_{pre}. Following Jia and Liang (2017), we add distracting sentences in the dialogue, which are similar to the *What* and *Who* questions. For example, assume a *Who* question is "Who is the girlfriend of Joey?". We construct a sentence "X is the girlfriend of Joey." where X is a random-select speaker, and randomly insert the distracting sentence into the dialogue. The distracting sentence is shown to confuse models using shallow patterns for reasoning. Results are shown in Table 5. From the results, the performance of BERT_{pre} drops seriously. By contrast, our approach demonstrates robustness in such adversarial testing scenarios. The reason is that our approach uses background knowledge for reasoning. With such background knowledge, our approach tends not to select the answer from these added distracting sentences, because these adversarial sentences do not have relationships with other parts of dialogue.

7 Ablation Study

7.1 Impact of Different Types of Knowledge

We compare the impact of different types knowledge on the results and we also investigate using system predicted results rather than golden knowledge (We train co-reference identifier and relation classifier following Zhou and Choi (2018) and Yu et al. (2020), resulting 74.4% in F1 (B^3) and 57.0%, matching the state-of-the-arts). The results are given in Table 6. From the results, relational knowledge is more effective than co-reference knowledge for this task. And their combination leads to the highest result. We note using system predicted knowledge for reasoning leads to a drop of performance, but still achieves better performance than SoTA (Li and Choi, 2020).

7.2 Impact of Graph Structure Modeling

Impact of Graph Architecture. We compare performance on different graph architectures, including Graph Convolution Networks (GCN) (Kipf and Welling, 2016) and Graph Attention Networks (GAT)

[4]https://www.dgl.ai/

Setting	Model	UM	SM	EM
NONE	BERT_{pre}	71.3	61.2	45.6
GOLD	BERT_{pre} + Coref	72.4	65.3	46.9
	BERT_{pre} + Rel	73.0	65.1	47.2
	BERT_{pre} + Coref + Rel	74.1	65.5	48.1
PREDICTED	BERT_{pre} + Coref	72.0 (↓0.4)	64.7 (↓0.6)	46.0 (↓0.9)
	BERT_{pre} + Rel	72.5 (↓0.5)	64.7 (↓0.4)	46.8 (↓0.4)
	BERT_{pre} + Coref + Rel	**73.4** (↓0.7)	**65.1** (↓0.4)	**47.4** (↓0.7)
	SoTA (Li and Choi, 2020)	73.3	63.1	46.8

Table 6: Results on the test set of FriendsQA by compared with existing models using the predicted relation/co-reference chain. GOLD and PREDICTED denote using the ground-truth annotated information or system predicted results to construct the relational graph for reasoning.

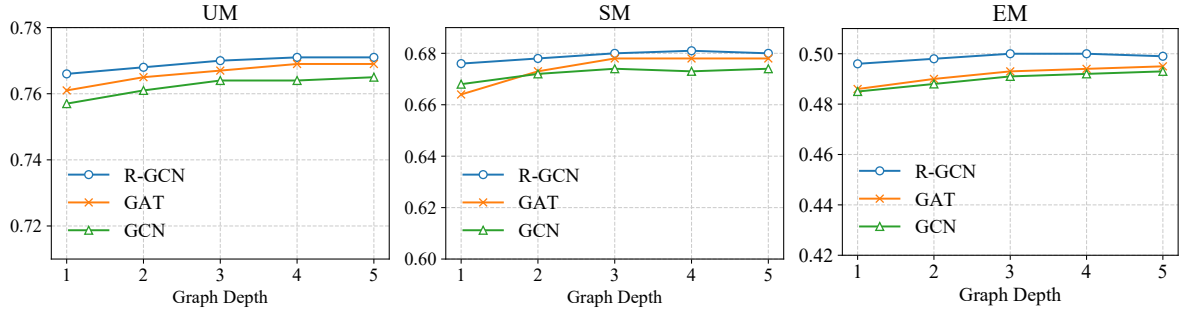

Figure 3: Results on the developing set of FriendsQA studying the impact of graph depth.

(Veličković et al., 2017). We also compare our model with a system simply concatenate the relational knowledge, in a triple format of (subject, predict, object), at the end of dialogue for reasoning. Results are shown in Table 7. From the results, our model, i.e., BERT_{pre} + Graph (R-GCN), achieves the best performance. The reason might be that GCN and GAT use the homogeneous graph to encode knowledge, where edges do not have type information, thus they cannot discern different types of knowledge. While R-GCN uses the heterogeneous graph, and uses different types of edges to indicate different types of relationships. Also note, simply concatenating knowledge leads to a negative result.

Impact of Graph Depth. Figure 3 shows the impact of the graph depth of R-GCN, GAT, and GCN. The results are on the development set. From the results, for GCN and GAT, more graph layers lead to better performance. While R-GCN is more effective, even with only one layer can it yield good performance.

8 Case Study

We study several cases by comparing the difference in predictions of BERT_{pre} and our approach. The results are shown in Table 8. (1) In the first example, Rach and Rachel Green have an *alternate:name* relation. And our model can integrate this knowledge for reasoning, which correctly predicts the answer for "What does Monica call Rachel for short?". By contrast, BERT_{pre} lacks the above knowledge. It wrongly outputs "my brother" as the answer. (2) In the second example, the question is "When does Dough come?", but the dialogue only contains "his boss approaches" in the scene utterance (A dialogue in FriendsQA can contain a special "scene utterance" describing backgrounds of a scene). To solve this problem, the model should figure out the relation between Doug and Chandler is *per:boss*. Our method can reason over such knowledge and find the correct answer. (3) In the third example, note BERT_{pre} incorrectly predicts Steven as the answer. While, Steven is actually the *per:alternative_names* of Stephen Waltham (and Steven and Stephen Waltham share a co-reference relation), which obviously can not be the answer of "Who did Stephen Waltham tell not to take that tone with him?". Our model is aware of *per:alternative_names* knowledge, and it will not make such mistake.

Model	UM	SM	EM
BERT_{pre}	71.3	61.2	45.6
BERT_{pre} + Linear	71.0	59.1	44.1
BERT_{pre} + Graph (GCN)	73.2	63.4	46.2
BERT_{pre} + Graph (GAT)	73.7	63.2	46.6
BERT_{pre} + Graph (R-GCN)	**74.1**	**66.9**	**48.1**

Table 7: Comparison of different methods to encode knowledge on the test set of FriendsQA. "+ Linear" indicates concatenating knowledge in the input sequence for reasoning; GCN, GAT, and R-GCN denote different graph architecture to encode background knowledge.

Monica Geller:	Uh, so, uh, **Rach**, uh ... do you wanna save this wrapping paper?
Rachel Green:	I don't know, I don't know.
	19 utterances are omitted ..
Monica Geller:	Then why the hell are you dumping my brother ?!?
Question:	What does Monica call Rachel for short ?
Ground-Truth:	**Rach**.
Predicted:	my brother (BERT_{pre}) ✗ | Rach (BERT_{pre} + Graph) ✓
[Scene: Chandler's office, **Chandler is bent over getting some water** as his boss approaches.]	
Doug:	Bing! Read your Computech proposal, a real homerun. Ooh. Barely got ya that time, get over here. Come on. Wham! Good one. That was a good one.
Chandler:	What is with him ?
	... other utterances are omitted ...
Question:	When does Dough come?
Ground-Truth:	**Chandler is bent over getting some water**.
Predicted:	Barely got ya (BERT_{pre}) ✗ | Chandler is ... some water (BERT_{pre} + Graph) ✓
Andrea Waltham:	This is ridiculous. I mean we had an agreement. Will you say something, Steven?! Please!!!
Stephen Waltham:	Don't take that tone with me. All right you can.
	... other utterances are omitted ...
Question:	Who did Stephen Waltham tell not to take that tone with him?
Ground-Truth:	**Andrea Waltham**.
Predicted:	Steven (BERT_{pre}) ✗ | Andrea Waltham (BERT_{pre} + Graph) ✓

Table 8: Results of case study comparing predictions of BERT_{pre} and our approach (BERT_{pre} + Graph). The answer to the question is annotated in *bold* in the dialogue.

9 Conclusion and Future Work

In this paper, we study the problem of question answering over dialog. We propose a new model that can effectively integrate background evidence for reasoning via a graph based knowledge integration process. The effectiveness of our approach is verified on extensive experiments. In the current study, we have used the results of additional co-reference identifier and relation extractor to build the relational graph, working in a pipeline style. In the future, we would study the inter-dependency of question answering and co-reference/relation identification tasks, in a multi-task setting to boost performance.

Acknowledgements

This work is supported by the National Key R & D Program of China (2020AAA0106400), the National Natural Science Foundation of China (No.61533018, No.61922085, No.61806201) and the Key Research Program of the Chinese Academy of Sciences (Grant NO. ZDBS-SSW-JSC006). This work is also supported by Beijing Academy of Artificial Intelligence(BAAI).

References

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium, October-November. Association for Computational Linguistics.

Henry Y. Chen, Ethan Zhou, and Jinho D. Choi. 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)*, pages 216–225, Vancouver, Canada, August. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. 2019. A lexicon-based graph neural network for Chinese NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1040–1050, Hong Kong, China, November. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend NIPS 2015. *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15)*, pages 1693–1701.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6.

Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

Changmao Li and Jinho D. Choi. 2020. Transformers to learn hierarchical contexts in multiparty dialogue for span-based question answering.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2019. Neural cross-lingual event detection with minimal parallel resources. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 738–748, Hong Kong, China, November. Association for Computational Linguistics.

Kaixin Ma, Tomasz Jurczyk, and Jinho D. Choi. 2018. Challenging reading comprehension on daily conversation: Passage completion on multiparty dialog. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2039–2048, New Orleans, Louisiana, June. Association for Computational Linguistics.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832, Melbourne, Australia, July. Association for Computational Linguistics.

Delai Qiu, Yuanzhe Zhang, Xinwei Feng, Xiangwen Liao, Wenbin Jiang, Yajuan Lyu, Kang Liu, and Jun Zhao. 2019. Machine reading comprehension using structural knowledge graph-aware network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5896–5901, Hong Kong, China, November. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, March.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10843 LNCS(1):593–607.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231, March.

Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. 2019. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3308–3318, Florence, Italy, July. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198, Vancouver, Canada, July. Association for Computational Linguistics.

Dirk Weissenborn. 2017. Reading twice for natural language understanding. *CoRR*, abs/1706.02596.

Zhengzhe Yang and Jinho D. Choi. 2019. FriendsQA: Open-domain question answering on TV show transcripts. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 188–197, Stockholm, Sweden, September. Association for Computational Linguistics.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in LSTMs for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446, Vancouver, Canada, July. Association for Computational Linguistics.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4927–4940, Online, July. Association for Computational Linguistics.

Ethan Zhou and Jinho D. Choi. 2018. They exist! introducing plural mentions to coreference resolution and entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 24–34, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.