Question-Interlocutor Scope Realized Graph Modeling over Key Utterances for Dialogue Reading Comprehension

Jiangnan Li[1,2,†,○], **Mo Yu**[3,†], **Fandong Meng**[3], **Zheng Lin**[1,2,*]
Peng Fu[1], **Weiping Wang**[1], **Jie Zhou**[3]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Pattern Recognition Center, WeChat AI, Tencent Inc.
{lijiangnan,linzheng,fupeng,wangweiping}@iie.ac.cn, moyumyu@global.tencent.com
{fandongmeng,withtomzhou}@tencent.com

Abstract

We focus on dialogue reading comprehension (DRC) that extracts answers from dialogues. Compared to standard RC tasks, DRC has raised challenges because of the complex speaker information and noisy dialogue context. Essentially, the challenges come from the speaker-centric nature of dialogue utterances — an utterance is usually insufficient in its surface form, but requires to incorporate the role of its speaker and the dialogue context to fill the latent pragmatic and intention information. We propose to deal with these problems in two folds. First, we propose a new key-utterances-extracting method, which can realize more answer-contained utterances. Second, based on the extracted utterances, we then propose a Question-Interlocutor Scope Realized Graph (QuISG). QuISG involves the question and question-mentioning speaker as nodes. To realize interlocutor scopes, utterances are connected with corresponding speakers in the dialogue. Experiments on the benchmarks show that our method achieves state-of-the-art performance against previous works.[1]

1 Introduction

Beyond the formal forms of text, dialogues are one of the most frequently used media that people communicate with others to informally deliver their emotions (Poria et al., 2019), opinions (Cox et al., 2020), and intentions (Qin et al., 2021). Moreover, dialogue is also a crucial information carrier in literature, such as novels and movies (Kociský et al., 2018), for people to understand the characters and plots (Sang et al., 2022) in their reading behaviors. Therefore, comprehending dialogues is a key step for machines to act like humans.

Despite the value of dialogues, reading comprehension over dialogues (DRC), which extracts answer spans for independent questions from dialogues, lags behind those of formal texts like news and Wikipedia articles.[2] The reason mainly comes from distinctive features of dialogues. Specifically, dialogues involve informal oral utterances which are usually short and incomplete, and thus understanding them highly depends on their loosely structured *dialogue context*. As a high-profile spot in the conversational-related domain, *dialogue context* modeling is also a major scientific problem in DRC.

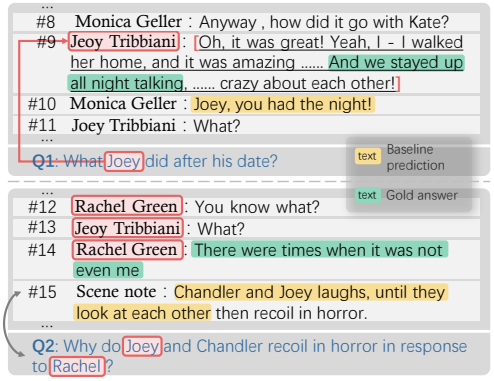

Figure 1: Two questions with related dialogue clips that the baseline SelfSuper (Li and Zhao, 2021) fails. Utter. #9 is too long, so we omit some parts of the utterance.

In previous works, Li and Zhao (2021) (abbreviated as SelfSuper) point out that *dialogue context* modeling in DRC faces two challenges: complex speaker information and noisy question-unrelated context. For speaker information, SelfSuper design a self-supervised task guessing who a randomly masked speaker is according to the dialogue context (e.g., masking "`Monica Geller`" of #10 in Fig. 1). To reduce noise, another task is made to predict whether an utterance contains the answer.

*Zheng Lin is the corresponding author. † Authors contributed equally to this work. ○ Joint work with Pattern Recognition Center, WeChat AI, Tencent Inc.

[1]https://github.com/LeqsNaN/QuISG

[2]Note there is a direction of conversational QA (Reddy et al., 2018; Choi et al., 2018; Sun et al., 2019) differing from DRC here. For the former, the Question-Answer process is formed as a dialogue, and the model derives answers from **Wikipedia articles or English exams**.

Findings of the Association for Computational Linguistics: ACL 2023, pages 4956–4968
July 9-14, 2023

Although decent performance can be achieved, several urging problems still exist.

Firstly, speaker guessing does not aware of the speaker information in questions and the interlocutor scope. As randomly masking is independent of the question, it cannot tell which speaker in the dialogue is related to the speaker mentioned in the question, e.g., `Joey Tribbiani` to `Joey` in Q1 of Fig. 1. As for the interlocutor scope, we define it as utterances said by the corresponding speaker. We point out that utterances have a speaker-centric nature: First, each utterance has target listeners. For example, in Utter. #10 of Fig. 1, it requires to understand that `Joey` is a listener, so "`you had the night`" is making fun of `Joey` from `Monica`'s scope. Second, an utterance reflects the message of the experience of its speaker. For example, to answer Q1 in Fig. 1, it requires understanding "`stayed up all night talking`" is the experience appearing in `Joey`'s scope. Due to ignoring the question-mentioned interlocutor and its scope, SelfSuper provides a wrong answer.

Secondly, answer-contained utterance (denoted as key utterance by SelfSuper) prediction prefers utterances similar to the question, failing to find key utterances not similar to the question. The reason is that answers are likely to appear in utterances similar to the question. For example, about 77% of questions have answers in top-5 utterances similar to the question according to SimCSE (Gao et al., 2021) in the dev set of FriendsQA (Yang and Choi, 2019). Furthermore, the utterances extracted by the key utterance prediction have over 82% overlaps with the top-5 utterances. Therefore, there are considerable key utterances have been ignored, leading to overrated attention to similar utterances, e.g., Q2 in Fig. 1. In fact, many key utterances are likely to appear near question-similar utterances because contiguous utterances in local contexts tend to be on one topic relevant to the question (Xing and Carenini, 2021; Jiang et al., 2023). However, the single utterance prediction cannot realize this.

To settle the aforementioned problems, so that more answer-contained utterances can be found and the answering process realizes the question and interlocutor scopes, we propose a new pipeline framework for DRC. We first propose a new key-utterances-extracting method. The method slides a window through the dialogue, where contiguous utterances in the window are regarded as a unit. The prediction is made on these units. Based on utterances in predicted units, we then propose Question-Interlocutor Scope Realized Graph (QuISG) modeling. QuISG constructs a graph over contextualized embeddings of words. The question and speaker names mentioned in the question are explicitly present in QuISG as nodes. To remind the model of interlocutor scopes, QuISG connects every speaker node in the dialogue with words from the speaker's scope. We verify our model on two popular DRC benchmarks. Our model achieves decent performance against baselines on both benchmarks, and further experiments indicate the efficacy of our method.

2 Related Work

Dialogue Reading Comprehension. Unlike traditional Machine Reading Comprehension (Rajpurkar et al., 2016), Dialogue Reading Comprehension (DRC) aims to answer a question according to the given dialogue. There are several related but different types of conversational question answering: CoQA (Reddy et al., 2018) conversationally asks questions after reading Wikipedia articles. QuAC (Choi et al., 2018) forms a dialogue of QA between a student and a teacher about Wikipedia articles. DREAM (Sun et al., 2019) tries to answer multi-choice questions over dialogues of English exams. These works form QA pairs as a conversation between humans and machines. To understand the characteristics of speakers, Sang et al. (2022) propose TVShowGuess in a multi-choice style to predict unknown speakers in dialogues.

Conversely, we focus on DRC extracting answer spans from a dialogue for an independent question (Yang and Choi, 2019). For DRC, Li and Choi (2020) propose several pretrained and downstream tasks on the utterance level. To consider the coreference of speakers and interpersonal relationships between speakers, Liu et al. (2020) introduce the two types of knowledge from other dialogue-related tasks and construct a graph to model them. Besides, Li et al. (2021); Ma et al. (2021) model the knowledge of discourse structure of utterances in the dialogues. To model the complex speaker information and noisy dialogue context, two self-supervised tasks, i.e., masked-speaker guessing and key utterance prediction, are utilized or enhanced by Li and Zhao (2021); Zhu et al. (2022); Yang et al. (2023). However, existing work ignores explicitly modeling the question and speaker scopes and suffers from low key-utterance coverage.

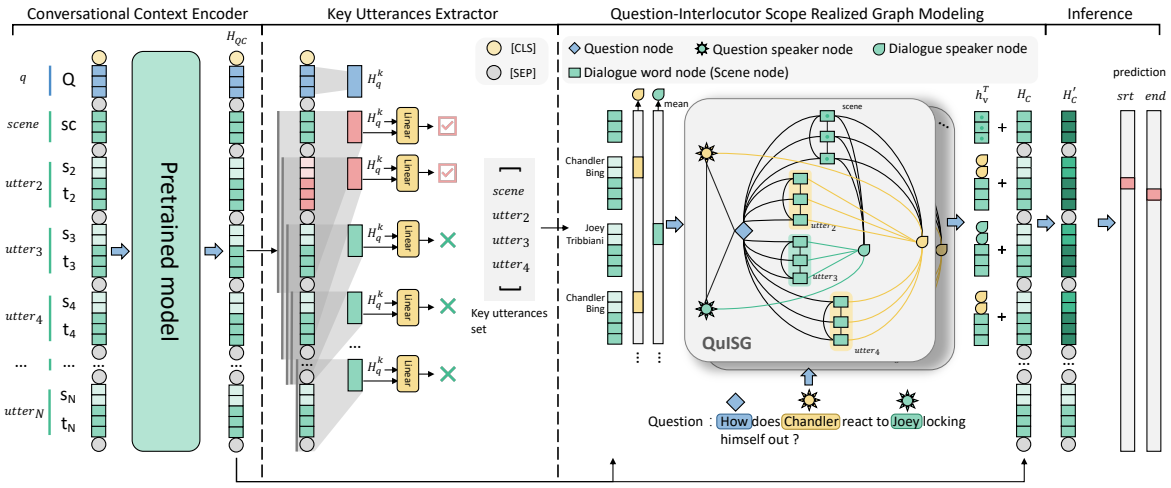

Figure 2: The overall framework of our proposed model. We first encode the dialogue and the question by pretrained models. The key utterances extractor takes contiguous utterances as a unit to extract key utterances. Based on extracted key utterances, the question-interlocutor scope realized graph is constructed.

Dialogue Modeling with Graph Representations. In many QA tasks (Yang et al., 2018; Talmor et al., 2019), graphs are the main carrier for reasoning (Qiu et al., 2019; Fang et al., 2020; Yasunaga et al., 2021). As for dialogue understanding, graphs are still a hotspot for various purposes. In dialogue emotion recognition, graphs are constructed to consider the interactions between different parties of speakers (Ghosal et al., 2019; Ishiwatari et al., 2020; Shen et al., 2021). In dialogue act classification, graphs model the cross-utterances and cross-tasks information (Qin et al., 2021). In dialogue semantic modeling, Bai et al. (2021) extend AMR (Banarescu et al., 2013) to construct graphs for dialogues. As for DRC, graphs are constructed for knowledge propagation between utterances by works (Liu et al., 2020; Li et al., 2021; Ma et al., 2021) mentioned above.

3 Framework

3.1 Task Definition

Given a dialogue consisting of N utterances: $\mathcal{D} = [utter_1, utter_2, ..., utter_N]$, the task aims to extract the answer span a for a question $q = [qw_1, qw_2, ..., qw_{L_q}]$ from $\mathcal{D}$, where qw_i is the i-*th* word in q and L_q is the length of q. In $\mathcal{D}$, each utterance $utter_i = \{\text{speaker} : s_i, \text{text} : t_i\}$ contains its corresponding speaker (e.g., s_i ="`Chandler Bing`") and text content $t_i = [tw_1, tw_2, ..., tw_{L_i}]$, where tw_j the j-*th* word in t_i and L_i is the length of t_i. For some unanswerable questions, there is no answer span to be found in $\mathcal{D}$. Under such a circumstance, a is assigned to be null.

3.2 Conversational Context Encoder

To encode words contextually using pretrained models (PTM), following previous work (Li and Zhao, 2021), we chronologically concatenate utterances in the same conversation to form a text sequence: $\mathcal{C}$ = "s_1: t_1 [SEP] ... [SEP] s_N: t_N" Holding the conversational context $\mathcal{C}$, PTM can deeply encode $\mathcal{C}$ with the question q to make it question-aware by concatenating them as QC = "[CLS] q [SEP] $\mathcal{C}$ [SEP]" (it is okay that $\mathcal{C}$ goes first). Following Li and Zhao (2021), we utilize the ELECTRA discriminator to encode the sequence QC:

$$H_{QC} = \text{ELECTRA}(QC), \quad (1)$$

where $H_{QC} \in \mathbb{R}^{L_{QC} \times d_h}$, L_{QC} is the length of QC, and d_h is the hidden size of PLM. H_{QC} can be split into $H_Q \in \mathbb{R}^{L_q \times d_h}$ and $H_C \in \mathbb{R}^{L_C \times d_h}$ according the position of [SEP] between q and C, where L_C is the length of $\mathcal{C}$.

3.3 Key Utterances Extractor

Treating every single utterance as a unit to pair with the question prefers utterances similar to the question. However, the utterance containing the answer is not always in the case, where it can appear near a similar utterance within several steps due to the high relevance of local dialogue topics. The key utterance extractor aims to extract more answer-contained utterances. We apply a window along the dialogue. Utterances in the window are treated as a unit so that the similar utterance and the answer-contained utterance can co-occur and more answer-contained utterances can be realized.

3.3.1 Training the Extractor

With the window whose size is m, [$utter_i$, $utter_{i+1}$, ..., $utter_{i+m}$] is grouped. Mapping the start (st_i) and end (ed_i) position of the unit in $\mathcal{C}$, the representation of the unit can be computed by:

$$H^k_{u_i} = \text{Maxpooling}(H_C[st_i : ed_i]). \quad (2)$$

Similarly, the representation of the question is computed by $H^k_q = \text{Maxpooling}(H_Q)$. The correlation score between them is then computed by:

$$y_i = \text{sigmoid}(\text{Linear}(H^k_{u_i} || H^k_q)), \quad (3)$$

where $\text{Linear}(\cdot)$ is a linear unit mapping the dimension from $\mathbb{R}^{2d_h}$ to $\mathbb{R}$. For the unit, if any utterances in it contain the answer, the label y^k_i of this unit is set to 1, otherwise 0. Therefore, the training objective of the key utterances extractor on the dialogue $\mathcal{D}$ is:

$$\mathcal{J}_k = -\sum_{i=1}^{N-m} [(1-y^k_i)log(1-y_i) + y^k_i log(y_i)]. \quad (4)$$

3.3.2 Extracting Key Utterances

The extractor predicts whether a unit is related to the question. If $y_i > 0.5$, the unit is regarded as a question-related unit, and utterances inside are all regarded as **key utterances**. To avoid involving too many utterances as key utterances, we rank all the units whose $y_i > 0.5$ and pick up top-k units. For a question q, we keep a key utterance set $key = (\cdot)$ to store the extracted key utterances.

Specifically, when the i-th unit satisfies the above condition, [$utter_i$, ..., $utter_{i+m}$] are all considered to be added into key. If $utter_i$ does not exist in key, then key.add($utter_i$) is triggered, otherwise skipped. After processing all the qualified units, key sorts key utterances by sort(key, 1→N), where 1→N denotes chronological order.

We obverse that, in most cases, key utterances in key are consecutive utterances. When k=3 and m=2, the set is ordered as ($utter_{i-m}$, ..., $utter_i$, ..., $utter_{i+m}$), where $utter_i$ is usually the similar utterance.

3.4 Question-Interlocutor Scope Realized Graph Modeling

To guide models to further realize the question, speakers in the question, and scopes of speakers in $\mathcal{D}$, we construct a Question-Interlocutor Scope Realized Graph (QuISG) based on key. QuISG is formulated as $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{A}$ denotes the adjacent matrix of edges. After the construction of QuISG, we utilize a node-type realized graph attention network to process it. We elaborate on QuISG below.

3.4.1 Nodes

We define several types of nodes for key utterances and the question.

Question Node: Question node denotes the questioning word (e.g., "what") of the question. The **node representation** is initialized by meanpooling the representations of the question words: v.rep=mean(H_Q[what]). We denote this **type of node** as v.t=qw.

Question Speaker Node: Considering speakers in the question can help models realize which speakers and their interactions are focused by the question. Question speaker node is derived from the speaker name recognized from the question. We use stanza (Qi et al., 2020)[3] performing NER to recognize person names (e.g. "ross") in the question and pick up those names appearing in the dialogue as interlocutors. Then, we have v.rep=H_Q[ross] and v.t=qs. Additionally, if a question contains no speaker name or the picked name does not belong to interlocutors in the dialogue, no question speaker node will be involved.

Dialogue Speaker Node: Speakers appearing in the dialogue are crucial for dialogue modeling. We construct speakers of key utterances as dialogue speaker nodes. As the speaker in the dialogue is identified by its full name (e.g., "Ross Gellar"), we compute the node embedding by meanpooling the full name and all key utterances of the speaker will provide its speaker name: v.rep=mean(H_C[Ross$_1$, Gellar$_1$, ..., Ross$_x$, Gellar$_x$]), where x is the number of key utterance whose speaker name is "Ross Gellar". We set v.t=ds.

Dialogue Word Node: As the main body to perform answer extraction, words from all key utterances are positioned in the graph as dialogue word nodes. The embedding is initialized from the corresponding item of H_C. This type is set to v.t=dw.

Scene Node: In some datasets, there is a kind of utterance that appears at the beginning of a dialogue and briefly describes the scene of the dialogue. If it is a key utterance, we set words in it as scene nodes. Although we define the scene node, it still acts as a dialogue word node with v.t=dw. The

[3] https://github.com/stanfordnlp/stanza

only difference is the way to connect with dialogue speaker nodes. We state it in Sec. 3.4.2.

3.4.2 Edges

Edges connect the defined nodes. The adjacent matrix of edges is initialized as $\mathcal{A} = \mathrm{O}$. As QuISG is an undirected graph, $\mathcal{A}$ is symmetric. We denote $\mathcal{A}[\mathrm{v}_x, \mathrm{v}_y] = 1$ as $\mathcal{A}[\mathrm{v}_1, \mathrm{v}_2] = 1$ and $\mathcal{A}[\mathrm{v}_2, \mathrm{v}_1] = 1$.

For the word node $\mathrm{v}_x \in utter_i$, we connect it with other word nodes $\mathrm{v}_y \in utter_i$ ($x - k_w \leq y \leq x + k_w$) within a window whose size is k_w, i.e., $\mathcal{A}[\mathrm{v}_x, \mathrm{v}_y] = 1$. For word nodes in other utterances (e.g., $\mathrm{v}_z \in utter_{i+1}$), no edge is set between v_x and v_z. To remind the model of **the scope of speakers**, we connect every word node with the *dialogue speaker node* v_{s_i} it belongs to, i.e., $\mathcal{A}[\mathrm{v}_x, \mathrm{v}_{s_i}] = 1$. To realize the question, we connect all word nodes with the *question node* v_q, i.e., $\mathcal{A}[\mathrm{v}_x, \mathrm{v}_q] = 1$.

For the speakers mentioned in the question, we fully connect their *question speaker nodes* to model interactions between these speakers, e.g., $\mathcal{A}[\mathrm{v}_{qs_m}, \mathrm{v}_{qs_n}] = 1$. To remind the model which speaker in dialogue is related, we connect the *question speaker node* v_{qs_m} with its *dialogue speaker node* v_{s_i}, i.e., $\mathcal{A}[\mathrm{v}_{qs_m}, \mathrm{v}_{s_i}] = 1$. Furthermore, *question speaker nodes* is connected with the *question node*, e.g., $\mathcal{A}[\mathrm{v}_{qs_m}, \mathrm{v}_q] = 1$.

If the scene description is selected as a key utterance, it will be regarded as an utterance without speaker identification. We treat a *scene node* as a word node and follow the same edge construction as word nodes. As the scene description may tell things about speakers, we utilize stanza to recognize speakers and connect all *scene nodes* with the corresponding *dialogue speaker nodes*.

For every node in QuISG, we additionally add a self-connected edge, i.e., $\mathcal{A}[\mathrm{v}, \mathrm{v}] = 1$.

3.4.3 Node-Type Realized Graph Attention Network

Node-Type Realized Graph Attention Network (GAT) is a T-layer stack of graph attention blocks (Velickovic et al., 2017). The input of GAT is a QuISG and GAT propagates and aggregates messages between nodes through edges.

We initial the graph representation by $h_v^0 = \mathrm{v.rep}$. A graph attention block mainly performs multi-head attention computing. We exemplify attention computing by one head. To measure how important the node v_n to the node v_m, the node type realized attentive weight is computed by:

$$\alpha_{mn} = \frac{\exp\left(\mathrm{LReLU}\left(\mathrm{c}_{mn}\right)\right)}{\sum_{\mathrm{v}_o \in \mathcal{N}_{\mathrm{v}_m}} \exp\left(\mathrm{LReLU}\left(\mathrm{c}_{mo}\right)\right)}, \quad (5)$$

$$\mathrm{c}_{mn} = \mathrm{a}\left[[h_{\mathrm{v}_m}^{t-1} || r_{\mathrm{v}_m.\mathrm{t}}]\mathrm{w}_q [h_{\mathrm{v}_n}^{t-1} || r_{\mathrm{v}_n.\mathrm{t}}]\mathrm{w}_k\right]^{\mathrm{T}}, \quad (6)$$

where $r_{\mathrm{v}_m.\mathrm{t}} \in \mathbb{R}^{1\times 4}$ is a one-hot vector denoting the node type of v_m, and $\mathrm{a} \in \mathbb{R}^{1\times 2d_{head}}$, $\mathrm{w}_q \in \mathbb{R}^{(d_{head}+4)\times d_{head}}$, $\mathrm{w}_k \in \mathbb{R}^{(d_{head}+4)\times d_{head}}$ are trainable parameters.

Furthermore, the graph attention block aggregates the weighted message by:

$$h_{\mathrm{v}_m}^{t\cdot head} = \mathrm{ELU}\left(\sum_{\mathrm{v}_o \in \mathcal{N}_{\mathrm{v}_m}} \alpha_{mn} h_{\mathrm{v}_o}^{t-1} \mathrm{W}_v\right), \quad (7)$$

where $\mathrm{W}_o \in \mathbb{R}^{d_{head}\times d_{head}}$ is a trainable parameter. By concatenating weighted messages from all heads, the t-th graph attention block can update the node representation from $h_{\mathrm{v}_m}^{t-1}$ to $h_{\mathrm{v}_m}^{t}$.

3.5 Answer Extraction

After graph modeling, nodes in the QuISG are then mapped back into the original token sequence. We locate the dialogue word (scene) node v_x to its corresponding token representation $H_C[utter_i[x]]$ in $\mathcal{C}$, and then update the token representation by $H_C[utter_i[x]]$ += $h_{\mathrm{v}_x}^T$. For the speaker token representation $H_C[\mathrm{Ross}_i, \mathrm{Gellar}_i]$ in key utterances, the mapped dialogue speaker node v_{s_i} updates it by $H_C[\mathrm{Ross}_i, \mathrm{Gellar}_i]$ += $[h_{\mathrm{v}_{s_i}}^T, h_{\mathrm{v}_{s_i}}^T]$. As a speaker name s_i may appear several times, we repeat adding $h_{\mathrm{v}_{s_i}}^T$ to the corresponding token representations. We denote the updated H_C as $H_C^{'}$.

3.5.1 Training

Given $H_C^{'}$, the model computes the start and the end distributions by:

$$Y_{srt} = \mathrm{softmax}(\mathrm{w}_{srt} {H_C^{'}}^{\mathrm{T}}), \quad (8)$$

$$Y_{end} = \mathrm{softmax}(\mathrm{w}_{end} {H_C^{'}}^{\mathrm{T}}), \quad (9)$$

where $\mathrm{w}_{srt} \in \mathbb{R}^{1\times L_C}$, $\mathrm{w}_{end} \in \mathbb{R}^{1\times L_C}$ are trainable parameters. For the answer span a, we denote its start index and end index as a_{st} and a_{ed}. Therefore, the answer extracting objective is:

$$\mathcal{J}_{ax} = -\log\left(Y_{srt}(a_{st})\right) - \log\left(Y_{end}(a_{ed})\right). \quad (10)$$

If there are questions without any answers, another header is applied to predict whether a question is answerable. The header computes the probability by $p_{na} = \mathrm{sigmoid}(\mathrm{Linear}(H_C^{'}[\mathrm{CLS}]))$. By

annotating every question with a label $q \in \{0, 1\}$ to indicate answerability, another objective is added: $\mathcal{J}_{na} = -[(1-q)\log(1-p_{na}) + q\log(p_{na})]$. In this way, the overall training objective is $\mathcal{J} = \mathcal{J}_{ax} + 0.5 * \mathcal{J}_{na}$.

3.5.2 Inference

Following Li and Zhao (2021), we extract the answer span by performing a beam search with the size of 5. We constrain the answer span in one utterance to avoid answers across utterances. To further emphasize the importance of key utterances, we construct a scaling vector $S \in \mathbb{R}^{1 \times L_C}$, where the token belonging to key utterances is kept with 1 and the token out of key utterances is assigned with a scale factor $0 \leq f \leq 1$. The scaling vector is multiplied on Y_{srt} and Y_{end} before softmax, and we then use the processed possibilities for inference.

4 Experimental Settings

Datasets. Following Li and Zhao (2021), we conduct experiments on **FriendsQA** (Yang and Choi, 2019) and **Molweni** (Li et al., 2020). As our work does not focus on unanswerable questions, we construct an answerable version of Molweni (**Molweni-A**) by removing all unanswerable questions. FriendsQA is an open-domain DRC dataset collected from TV series. It contains 977/122/123 (train/dev/test) dialogues and 8,535/1,010/1,065 questions. Recognizing person names in questions, we find about 76%/76%/75% of questions contain person names in FriendsQA. Molweni is another dataset with topics on Ubuntu. It contains 8,771/883/100 dialogues and 24,682/2,513/2,871 questions, in which about 14% of questions are unanswerable. Dialogues in Mowelni are much shorter than in FriendsQA and contain no scene descriptions. Speaker names in Molweni are meaningless user ids (e.g., "nbx909"). Furthermore, questions containing user ids in Molweni, whose proportion is about 47%/49%/48%, are less than FriendsQA. In Molweni-A, there are 20,873/2,346/2,560 questions.

Compared Methods. We compare our method with existing methods in DRC. **ULM+UOP** (Li and Choi, 2020) adapt several utterance-level tasks to pretrain and finetune BERT in the multitask setting. **KnowledgeGraph** (Liu et al., 2020) introduces and structurally models additional knowledge about speakers' co-reference and social relations from other related datasets (Yu et al., 2020).

	Model	EM	F1
BERT based	ULM+UOP (Li and Choi, 2020)	46.80	63.10
	KnowledgeGraph (Liu et al., 2020)	46.40	64.30
	SelfSuper (Li and Zhao, 2021)	46.90	63.90
ELECTRA based	Reimpl. ELECTRA	54.62	71.29
	Reimpl. EKIM (Zhu et al., 2022)	56.45	72.45
	SelfSuper (Li and Zhao, 2021)	55.80	72.30
	Ours	**57.79***	**75.22***

Table 1: Results on FriendsQA. * denotes significance against SelfSuper with the t-test.

DADGraph (Li et al., 2021) is another graph-based method that introduces external knowledge about the discourse structure of dialogues. **ELECTRA** (Clark et al., 2020) is a vanilla fine-tuned ELECTRA. **SelfSuper** (Li and Zhao, 2021) is the SOTA method. It designs two self-supervised tasks to capture speaker information and reduce noise in the dialogue. **EKIM** (Zhu et al., 2022) is the Enhanced Key-utterance Interactive Model, which can be regarded as an enhanced SelfSuper with additional bi-attention to model the interaction between context, question, and key utterance. We reimplement EKIM in our experimental environment.

Implementation. Our model is implemented based on `ELECTRA-large-discriminator` from *Transformers*. For key utterances extraction, the size of the window (i.e., m) is set to 2 and top-3 units are considered. Other hyper-parameters are the same as those in the question-answering training. For question answering, we search the size of the word node window (i.e., k_w) in 1, 2, 3, and the number of attention heads in 1, 2, 4. We set the number of GAT layers to 5 for FriendsQA and 3 for Molweni; f is set to 0.5 for FriendsQA and 0.9 for Molweni. Other hyper-parameters are in Appendix A. We use the Exact Matching (EM) score and F1 score as the metrics.

5 Results and Discussion

5.1 Main Results

Tab. 1 shows the results achieved by our method and other baselines on FriendsQA. The baselines listed in the first three rows are all based on BERT. We can see that SelfSuper achieves better or competitive results compared with ULM+UOP and KnowledgeGraph This indicates the effectiveness of the self-supervised tasks for speaker and key utterance modeling of SelfSuper. When it comes to ELECTRA, the performance reaches a new ele-

	Model	EM	F1
BERT based	DADGraph (Li et al., 2021)	46.50	61.50
	SelfSuper (Li and Zhao, 2021)	49.20	64.00
ELECTRA based	Our Reimpl. ELECTRA	57.85	72.17
	Reimpl. EKIM (Zhu et al., 2022)	57.85	**72.95**
	SelfSuper (Li and Zhao, 2021)	58.00	**72.90**
	Ours	**59.32***	**72.86**
	Human performance	64.30	80.20

Table 2: Results on Molweni.

Model	EM	F1
ELECTRA	61.02	77.62
EKIM (Zhu et al., 2022)	61.76	78.26
SelfSuper (Li and Zhao, 2021)	61.13	78.30
Ours	**62.54***	**78.65**

Table 3: Results on Molweni-A.

vated level, which shows that ELECTRA is more suitable for DRC. By comparing with SelfSuper and EKIM, our method can achieve significantly better performance. This improvement shows the advantage of both the higher coverage of answer-contained utterances by our method and better graph representations to consider the question and interlocutor scopes by QuISG.

Results on Molweni are listed in Tab. 2. Our approach still gives new state-of-the-art, especially a significant improvement in EM scores. However, the absolute improvement is smaller compared to that of FriendsQA. This is mainly for two reasons. First, the baseline results are close to the human performance on Molweni, so the space for improvement is smaller. Second, Molweni contains unanswerable questions, which are not the main focus of our work. To see how the unanswerable questions affect the results, we further show the performance of our method and baselines on Molweni-A in Tab. 3, i.e., the subset of Molweni with only answerable questions. We observe that our method still achieves a better EM score against baselines and gains a slightly better F1 score, which indicates that our method can better deal with questions with answers. As for unanswerable questions, we believe that better performance can be achieved with related techniques plugged into our method, which we leave to future work.

By comparing the performance of our method in FriendsQA and Molweni, we can observe that our method is more significant in FriendsQA. We think the reason may be that (1) our key utterance extractor can cover more answer-contained utterances in FriendsQA, as will be shown in Fig. 3; (2) questions mentioning speakers show more frequently in FriendsQA than in Molweni, and therefore QuISG can help achieve better graph representations in FriendsQA. On all accounts, this further demonstrates that our method alleviates the problems that we focus on.

Model	FriendsQA		Molweni	
	EM	F1	EM	F1
full model	**57.79**	**75.22**	**59.32**	**72.86**
w/o NodeType	56.79	74.01	58.38	72.75
w/o KeyUttExt	55.87	72.30	58.48	72.10
w/o Q	56.37	73.55	58.20	72.52
w/o SpkScope	57.29	74.26	58.62	72.29
w/o All	53.12	70.05	56.32	71.08

Table 4: Ablation Study.

5.2 Ablation Study

To demonstrate the importance of our proposed modules, we adapt an ablation study. The results are shown in Tab. 4. We study the effects of node type information (NodeType), key utterances extractor and its scaling factor on logits (KeyUttExt); question and question speaker nodes (Q); edges between dialogue word nodes and dialogue speaker nodes to model interlocutor scope (SpkScope). We further remove both KeyUttExt and QuISG, leading to full connections between every two tokens in dialogues, and apply transformer layers to further process dialogues (w/o All).

By removing NodeType, the performance drops, which demonstrates minding different node behaviors can help better model graph representations. Our method w/o KeyUttExt decreases the performance, which demonstrates that the key utterance extractor is a crucial module for our method to find more answer-contained utterances and guides our model to pay more attention to the key part in a dialogue. As for the model w/o KeyUttExt shows more performance drop in FriendsQA, we think the reason may be that dialogues in FriendsQA are much longer than Molweni. Therefore, KeyUttExt can reduce more question-unrelated parts of dialogues for further graph modeling in FriendsQA. Removing Q or SpkScope also shows a performance decline, which indicates the importance of realizing the question and interlocutor scopes. Replacing KeyUttExt and QuISG with transformer layers even performs worse than ELECTRA, which

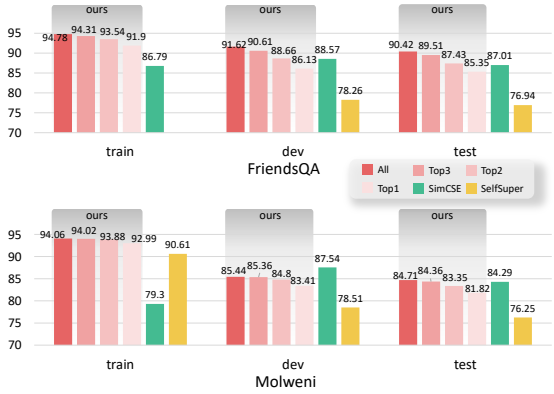

Figure 3: Recall of answer coverage using different key utterance extracting methods.

indicates that the further process of dialogues without speaker and question-realized modeling is redundant.

5.3 Accuracy of Utterance Extraction

As we claim that our method covers more answer-contained utterances compared with SelfSuper (EKIM has a similar result as SelfSuper), in this section, we show the recall of answer-contained utterances by different methods. Besides our method and SelfSuper, we further consider retrieval methods appearing in other reading comprehension tasks. As the similarity-based seeker is usually used, we apply the SOTA model SimCSE (Gao et al., 2021) to compute the similarity between utterances and the question. However, directly using top similar utterances produces an extremely low recall. Therefore, we also add utterances around every picked top utterance as key utterances like ours. We consider top 3 similar utterances and 4 context utterances around them. The results are illustrated in Fig. 3. As shown in Fig. 3, we choosing top 3 units for our key utterances does not affect the recall a lot and can keep the average size of key utterance set to 4.13 for FriendsQA and 3.61 for Molweni[4]. Compared with our method, SelfSuper achieves undesirable recall for answer-contained utterance extraction, which indicates the efficacy of our method. As for SimCSE, equipped with our enhancement, it can achieve competitive recall to ours. Especially on the dev and test sets of Molweni. However, the average size of the key utterance set of SimCSE is 7.73, whereas the average length of dialogue in Molweni is 8.82. Additionally, SimCSE extracts key utterances for every question

[4]The average length of dialogues in FriendsQA is 21.92 and 7.73 in Molweni.

Method	FriendsQA		Molweni	
	EM	F1	EM	F1
ours	57.79	75.22	59.32	72.86
SimCSE	57.19	74.36	57.30	72.09

Table 5: Results of our method and the variant with SimCSE searching for key utterances.

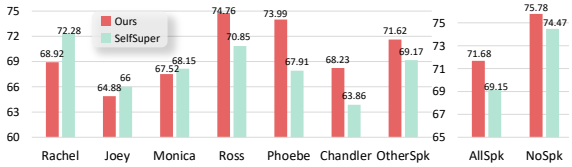

Figure 4: F1 scores of answers to the questions with or without speaker names in the dev set of FriendsQA.

regardless of the answerability, leading to a low recall for the Molweni training set.

To further show that our method is more suitable for DRC against SimCSE, we run a variant with key utterances extracted by SimCSE. The results are shown in Tab. 5. Our method achieves better performance with high coverage of answer-contained utterances and fewer key utterances.

5.4 Improvement on Questions with Speakers

As QuISG focuses on the question speaker information and dialogue interlocutor scope modeling, whether it can help answer questions mentioning speaker names is crucial to verify. We illustrate F1 scores of questions containing different speakers in FriendsQA and questions with or without mentioning speakers in Fig. 4. We can see that SelfSuper outperforms our method only on "`Rachel`" and is slightly better on "`Joey`" and "`Monica`". Our method can outperform SelfSuper by a great margin on "`Ross`", "`Phoebe`", "`Chandler`", and other casts. Furthermore, our method can improve the F1 score of speaker-contained questions by a wider margin compared to questions without speakers. This indicates that our speaker modeling benefits from our proposed method.

5.5 Case Study

At the very beginning of the paper, Fig. 1 provides two cases in that SelfSuper fails. On the contrary, attributing to our proposed key utterances extractor and QuISG, our method can answer the two questions correctly.

6 Conclusion

To cover more key utterances and make the model realize speaker information in the question and interlocutor scopes in the dialogue for DRC, we propose a new pipeline method. The method firstly adapts a new key utterances extractor with contiguous utterances as a unit for prediction. Based on utterances of the extracted units, a Question-Interlocutor Scope Realized Graph (QuISG) is constructed. QuISG sets question-mentioning speakers as question speaker nodes and connects the speaker node in the dialogue with words from its scope. Our proposed method achieves decent performance on related benchmarks.

Limitation

As our method does not focus on dealing with unanswerable questions, our method may not show a great advantage over other methods when there are a lot of unanswerable questions. How to improve the recognition of this type of question, avoid overrating further modeling on them, and therefore give more accurate graph modeling on answerable questions will be left to our future work. Besides, our speaker modeling prefers questions focusing on speakers, and it may show limited improvement if a dataset contains few speaker-related questions. However, speakers are key roles in dialogues, and therefore, questions about speakers naturally appear frequently in DRC.

The power of our key utterance extraction method to other QA fields remains unknown. It can be future work to extend it to other reading comprehension tasks like NarrativeQA (Kociský et al., 2018).

Our method does not involve additional knowledge, such as speakers' co-reference and relations (Liu et al., 2020), discourse structures of dialogues (Li et al., 2021; Ma et al., 2021), and decoupled bidirectional information in dialogues (Li et al., 2022). These types of knowledge, which are orthogonal to our work, are key components of dialogues. Therefore, making full use of the additional knowledge in dialogues with our graph modeling can be an interesting direction to explore.

Acknowledgement

This work was supported by National Natural Science Foundation of China (No. 61976207).

References

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic representation for dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4430–4445. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *LAW-ID@ACL*, pages 178–186. The Association for Computer Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2174–2184. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ramon Alfonso Villa Cox, Sumeet Kumar, Matthew Babcock, and Kathleen M. Carley. 2020. Stance in replies and quotes (SRQ): A new dataset for learning stance in twitter conversations. *CoRR*, abs/2006.00691.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8823–8838. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander F. Gelbukh. 2019. Dialoguegcn: A graph convolutional neural network for emotion recognition in conversation. 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, Hong Kong, China,*

November 3-7, 2019, pages 154–164. Association for Computational Linguistics.

Taichi Ishiwatari, Yuki Yasuda, Taro Miyazaki, and Jun Goto. 2020. Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7360–7370. Association for Computational Linguistics.

Junfeng Jiang, Chengzhang Dong, Akiko Aizawa, and Sadao Kurohashi. 2023. Superdialseg: A large-scale dataset for supervised dialogue segmentation. *CoRR*, abs/2305.08371.

Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Trans. Assoc. Comput. Linguistics*, 6:317–328.

Changmao Li and Jinho D. Choi. 2020. Transformers to learn hierarchical contexts in multiparty dialogue for span-based question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5709–5714. Association for Computational Linguistics.

Jiaqi Li, Ming Liu, Min-Yen Kan, Zihao Zheng, Zekun Wang, Wenqiang Lei, Ting Liu, and Bing Qin. 2020. Molweni: A challenge multiparty dialogues-based machine reading comprehension dataset with discourse structure. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2642–2652. International Committee on Computational Linguistics.

Jiaqi Li, Ming Liu, Zihao Zheng, Heng Zhang, Bing Qin, Min-Yen Kan, and Ting Liu. 2021. Dadgraph: A discourse-aware dialogue graph neural network for multiparty dialogue machine reading comprehension. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pages 1–8. IEEE.

Yiyang Li and Hai Zhao. 2021. Self- and pseudo-self-supervised prediction of speaker and key-utterance for multi-party dialogue reading comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 2053–2063. Association for Computational Linguistics.

Yiyang Li, Hai Zhao, and Zhuosheng Zhang. 2022. Back to the future: Bidirectional information decoupling network for multi-turn dialogue modeling. *CoRR*, abs/2204.08152.

Jian Liu, Dianbo Sui, Kang Liu, and Jun Zhao. 2020. Graph-based knowledge integration for question answering over dialogue. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2425–2435. International Committee on Computational Linguistics.

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2021. Enhanced speaker-aware multi-party multi-turn dialogue comprehension. *CoRR*, abs/2109.04066.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 527–536. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 101–108. Association for Computational Linguistics.

Libo Qin, Zhouyang Li, Wanxiang Che, Minheng Ni, and Ting Liu. 2021. Co-gat: A co-interactive graph attention network for joint dialog act recognition and sentiment classification. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13709–13717. AAAI Press.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Proceedings of the 57th conference of the association for computational linguistics, ACL 2019, florence, italy, july 28-august 2, 2019, volume 1: Long papers. In *ACL*, pages 6140–6150. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. Coqa: A conversational question answering challenge. *CoRR*, abs/1808.07042.

Yisi Sang, Xiangyang Mou, Mo Yu, Shunyu Yao, Jing Li, and Jeffrey Stanton. 2022. Tvshowguess: Character comprehension in stories as speaker guessing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*

NAACL 2022, Seattle, WA, United States, July 10-15, 2022, pages 4267–4287. Association for Computational Linguistics.

Weizhou Shen, Siyue Wu, Yunyi Yang, and Xiaojun Quan. 2021. Directed acyclic graph network for conversational emotion recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1551–1560. Association for Computational Linguistics.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge dataset and models for dialogue-based reading comprehension. *Trans. Assoc. Comput. Linguistics*, 7:217–231.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158. Association for Computational Linguistics.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *CoRR*, abs/1710.10903.

Linzi Xing and Giuseppe Carenini. 2021. Improving unsupervised dialogue topic segmentation with utterance-pair coherence scoring. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2021, Singapore and Online, July 29-31, 2021*, pages 167–177. Association for Computational Linguistics.

Tianqing Yang, Tao Wu, Song Gao, and Jingzong Yang. 2023. Dialogue logic aware and key utterance decoupling model for multi-party dialogue reading comprehension. *IEEE Access*, 11:10985–10994.

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, SIGdial 2019, Stockholm, Sweden, September 11-13, 2019*, pages 188–197. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 535–546. 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, ACL 2020, Online, July 5-10, 2020*, pages 4927–4940. Association for Computational Linguistics.

Xingyu Zhu, Jin Wang, and Xuejie Zhang. 2022. An enhanced key-utterance interactive model with decoupled auxiliary tasks for multi-party dialogue reading comprehension. In *International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022*, pages 1–8.

A Computation Resource and Other Setup

We use a piece of NVIDIA GeForce 3090 whose memory size is 24GB. All experiments require memory that is not more than 24GB. It takes 10-25 minutes for our model to finish an epoch training.

As for other hyperparameters used in our experiment, we follow Li and Zhao (2021) to set the learning rate to 4e-6 for FriendsQA and search learning rate from [1.4e-5, 1.2e-5, 1e-5, 8e-6] for Molweni (Molweni-A). The batch size is set to 4 for FriendsQA and 8 for Molweni (Molweni-A). The number of epochs is set to 3 for FriendsQA and 5 for Molweni (Molweni-A). The evaluation is made every 1/5 epoch for FriendsQA and 1/2 epoch for Molweni (Molweni-A). For GAT, the dropout is set to 0.1. During the training process, the learning rate linearly warms up with the portion of 0.01 to all steps and then linearly decays to zero. AdamW with adam epsilon of 1e-6 is utilized as the optimizer. 4 runs are adapted and the max one is picked.

For the utilization of SimCSE, *Transformers* version of `sup-simcse-roberta-large`, which achieves the best performance among all SimCSE variants on Avg. STS, is picked.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Sec. 7

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abs, Sec. 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Sec. 3.4.1, Sec. 4

☑ B1. Did you cite the creators of artifacts you used?
Sec. 3.4.1, Sec. 4

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Sec. 4

C ☑ Did you run computational experiments?

Appendix A

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Sec. 4, Appendix A

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Sec.4, Appendix A

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Sec.3.4.1

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.