GIFT: Graph-Induced Fine-Tuning for Multi-Party Conversation Understanding

Jia-Chen Gu[1], **Zhen-Hua Ling**[1,*], **Quan Liu**[2,3], **Cong Liu**[1,3], **Guoping Hu**[2,3]
[1]National Engineering Research Center of Speech and Language Information Processing,
University of Science and Technology of China, Hefei, China
[2]State Key Laboratory of Cognitive Intelligence [3]iFLYTEK Research, Hefei, China
{gujc,zhling}@ustc.edu.cn, {quanliu,congliu2,gphu}@iflytek.com

Abstract

Addressing the issues of *who* saying *what* to *whom* in multi-party conversations (MPCs) has recently attracted a lot of research attention. However, existing methods on MPC understanding typically embed interlocutors and utterances into sequential information flows, or utilize only the superficial of inherent graph structures in MPCs. To this end, we present a plug-and-play and lightweight method named **g**raph-**i**nduced **f**ine-**t**uning (GIFT) which can adapt various Transformer-based pre-trained language models (PLMs) for universal MPC understanding. In detail, the full and equivalent connections among utterances in regular Transformer ignore the sparse but distinctive dependency of an utterance on another in MPCs. To distinguish different relationships between utterances, four types of edges are designed to integrate graph-induced signals into attention mechanisms to refine PLMs originally designed for processing sequential texts. We evaluate GIFT by implementing it into three PLMs, and test the performance on three downstream tasks including addressee recognition, speaker identification and response selection. Experimental results show that GIFT can significantly improve the performance of three PLMs on three downstream tasks and two benchmarks with only 4 additional parameters per encoding layer, achieving new state-of-the-art performance on MPC understanding.

1 Introduction

Maintaining appropriate human-computer conversation is an important task leaping towards advanced artificial intelligence. Most of existing methods have studied understanding conversations between two participants, aiming at returning an appropriate response either in a generation-based (Shang et al., 2015; Serban et al., 2016; Zhang et al., 2020; Roller et al., 2021) or retrieval-based manner (Wu et al., 2017; Zhou et al., 2018; Tao et al., 2019; Gu et al., 2020). Recently, researchers have paid more attention to a more practical and challenging scenario involving more than two participants, which is well known as multi-party conversations (MPCs) (Ouchi and Tsuboi, 2016; Zhang et al., 2018; Le et al., 2019; Hu et al., 2019; Wang et al., 2020; Gu et al., 2021, 2022). Unlike two-party conversations, utterances in an MPC can be spoken by anyone and address anyone else in this conversation, constituting a *graphical* information flow and various relationships between utterances as shown in Figure 1(a). Thus, predicting who the next speaker will be (Meng et al., 2018) and who the addressee of an utterance is (Ouchi and Tsuboi, 2016; Zhang et al., 2018; Le et al., 2019) are unique and important issues in MPCs.

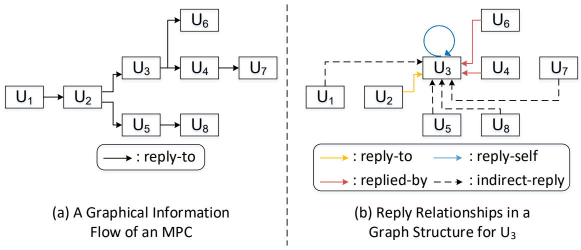


Figure 1: Illustration of (a) a graphical information flow of an MPC where rectangles denote utterances, and solid lines represent the "*reply*" relationship between two utterances, and (b) the detailed reply relationships between each utterance and U_3.

The complicated interactions between interlocutors, between utterances and between an interlocutor and an utterance naturally increase the difficulty of fully understanding MPCs. Existing studies on MPC understanding focus on the challenging issue of modeling the complicated conversation structures and information flows. The current state-of-the-art method MPC-BERT (Gu et al., 2021) proposed to pre-train a language model with two types of self-supervised tasks for modeling interlocutor structures and utterance semantics respectively in a unified framework. The complementary structural

*Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics Volume 1: Long Papers, pages 11645–11658
July 9-14, 2023 ©2023 Association for Computational Linguistics

and semantic information in MPCs is learned by designing a variety of self-supervised optimization objectives. However, the semantics contained in the interlocutor and utterance representations may not be effectively captured as these supervision signals are placed only on top of language models. During encoding inside language models, the full and equivalent connections among utterances in regular Transformer (Vaswani et al., 2017) ignore the sparse but distinctive dependency of an utterance on another, such as "*reply-to*". Despite of the performance improvement with pre-training, MPC-BERT still overlooks the inherent MPC graph structure when fine-tuning on various downstream tasks. Intuitively, leveraging graph-induced signals when fine-tuning pre-trained language models (PLMs) may yield better contextualized representations of interlocutors and utterances and enhance conversation understanding, but has been overlooked in previous studies.

In light of the above issues, we propose a plug-and-play and lightweight method named **g**raph-**i**nduced **f**ine-**t**uning (GIFT), which can adapt various Transformer-based PLMs and improve their ability for universal MPC understanding. Existing Transformer-based PLMs such as BERT (Devlin et al., 2019) are originally designed for processing sequential texts. To distinguish different relationships between utterances, four types of edges (reply-to, replied-by, reply-self and indirect-reply) are designed to integrate graph-induced signals in the attention mechanism. These edge-type-dependent parameters are utilized to refine the attention weights and to help construct the graphical conversation structure in Transformer. Intuitively, the conversation structure influences the information flow in MPCs, thus it can be used to strengthen the representations of utterance semantics. By this means, it can help characterize fine-grained interactions during the internal encoding of PLMs, and produce better representations that can be effectively generalized to multiple downstream tasks of MPCs. Lastly, the proposed method is plug-and-play which can be implemented into various Transformer-based PLMs, and is lightweight which requires only 4 additional parameters per encoding layer.

To measure the effectiveness of the proposed GIFT method and to test its generalization ability, GIFT is implemented into three PLMs including BERT (Devlin et al., 2019), SA-BERT (Gu et al., 2020) and MPC-BERT (Gu et al., 2021). We evaluate the performance on three downstream tasks including *addressee recognition, speaker identification* and *response selection*, which are three core research issues of MPCs. Two benchmarks based on Ubuntu IRC channel are employed for evaluation. One was released by Hu et al. (2019). The other was released by Ouchi and Tsuboi (2016) with three experimental settings according to session lengths. Experimental results show that GIFT helps improve the performance of all three PLMs on all three downstream tasks. Take MPC-BERT as an example, GIFT improved the performance by margins of 0.64%, 1.64%, 3.46% and 4.63% on the test sets of these two benchmarks respectively in terms of utterance precision of addressee recognition, by margins of 6.96%, 23.05%, 23.12% and 22.99% respectively in terms of utterance precision of speaker identification, and by margins of 1.76%, 0.88%, 2.15% and 2.44% respectively in terms of response recall of response selection, achieving new state-of-the-art performance on MPC understanding.

In summary, our contributions in this paper are three-fold: (1) A **g**raph-**i**nduced **f**ine-**t**uning (GIFT) method is proposed to construct and to utilize the inherent graph structure for MPC understanding. (2) GIFT is implemented into three PLMs and is tested on three downstream tasks to comprehensively evaluate the effectiveness and generalization ability. (3) The proposed method achieves new state-of-the-art performance on three downstream tasks and two benchmarks.

2 Related Work

Existing methods on building dialogue systems can be generally categorized into studying two-party conversations and multi-party conversations (MPCs). In this paper, we study MPCs. In addition to predicting the utterance, the tasks of identifying the *speaker* and recognizing the *addressee* of an utterance are also important for MPCs. Ouchi and Tsuboi (2016) first proposed the task of addressee and response selection and created an MPC corpus for studying this task. Zhang et al. (2018) proposed the speaker interaction RNN, which updated the speaker embeddings role-sensitively for addressee and response selection. Meng et al. (2018) proposed a task of speaker classification as a surrogate task for general speaker modeling. Le et al. (2019) proposed a who-to-whom (W2W) model to recognize the addressees of all utterances in an MPC. Kummerfeld et al. (2019) created a dataset based on Ubuntu IRC channel which was manually annotated with reply-structure graphs for MPC disentanglement. Hu et al. (2019) proposed a graph-structured neural network (GSN), the core of which is to encode utterances based on the graph topology rather than the sequence of their appearances to model the information flow as graphical. Wang et al. (2020) proposed to track the dynamic topic for response selection. Liu et al. (2020, 2021) studied transition-based online MPC disentanglement by modeling semantic coherence within each session and exploring unsupervised co-training through reinforcement learning. Gu et al. (2021) proposed MPC-BERT pre-trained with two types of self-supervised tasks for modeling interlocutor structures and utterance semantics. Gu et al. (2022) proposed HeterMPC to model the complicated interactions between utterances and interlocutors with a heterogeneous graph.

Compared with MPC-BERT (Gu et al., 2021) that is the most relevant to this work, two main differences should be highlighted. First, MPC-BERT works on designing various self-supervised tasks for pre-training, while GIFT works on further improving fine-tuning performance. Second, MPC-BERT models conversation graph structures by placing self-supervision signals on top of PLMs, while GIFT achieves this by alternatively modifying the internal encoding of PLMs. Furthermore, compared with GSN (Hu et al., 2019) and HeterMPC (Gu et al., 2022) that both attempt to model graphical information flows, it should be noted that there are also two main differences. First, GSN and HeterMPC represent each individual utterance as a node vector encoded by either BiLSTM (Hochreiter and Schmidhuber, 1997) or Transformer (Vaswani et al., 2017), and then update via graph neural network-based information passing, while this work integrates graph-induced signals into the fully-connected interactions of Transformer over the whole MPC context. Second, GSN and HeterMPC are designed specifically for MPC response generation, while this work focuses on universal MPC understanding. Overall, to the best of our knowledge, this paper makes the first attempt to design a fine-tuning method that leverages graph-induced signals during the internal encoding of Transformer-based PLMs for improving MPC understanding.

3 Graph-Induced Fine-Tuning (GIFT)

An MPC instance is composed of a sequence of (*speaker, utterance, addressee*) triples, denoted as $\{(s_n, u_n, a_n)\}_{n=1}^{N}$, where N is the number of turns in the conversation. Our goal is to fine-tune PLMs for universal MPC understanding. Given an MPC, it is expected to produce embedding vectors for all utterances which contain not only the semantic information of each utterance, but also the speaker and addressee structure of the whole conversation. Thus, it can be effectively adapted to various tasks by fine-tuning model parameters.

3.1 Intuition

Graphs are ubiquitous data structures. There is a wide range of application domains where data can be represented as graphs. For learning on graphs, graph neural networks (GNNs) (Scarselli et al., 2009) have emerged as the most powerful tool in deep learning. In short, GNNs take in a graph with node and edge features, and build abstract feature representations of nodes and edges by taking the available explicit connectivity structure (i.e., graph structure) into account. The so-generated features are then passed to downstream classification layers.

In this work, an MPC is viewed as a conversation graph. The current state-of-the-art method MPC-BERT (Gu et al., 2021) concatenates all utterances into a sequential text and sends it into Transformer-based PLMs for encoding. Recently, Transformer-based neural networks have been proven effective for representation learning and on a wide range of applications in natural language processing (NLP) such as machine translation (Vaswani et al., 2017) and language modeling (Devlin et al., 2019). Since Transformer considers full attention while building contextualized word representations, the full and equivalent connections among utterances ignore the sparse but distinctive dependency of an utterance on another. More importantly, recent studies on MPCs have indicated that the complicated graph structures can provide crucial interlocutor and utterance semantics (Hu et al., 2019; Gu et al., 2022). Thus, it inspires us to refine Transformer-based PLMs by modeling graph structures during internal encoding to help enhance the conversation understanding process.

3.2 Input Representation

Following Gu et al. (2020) and Gu et al. (2021), another type of speaker embeddings is added to

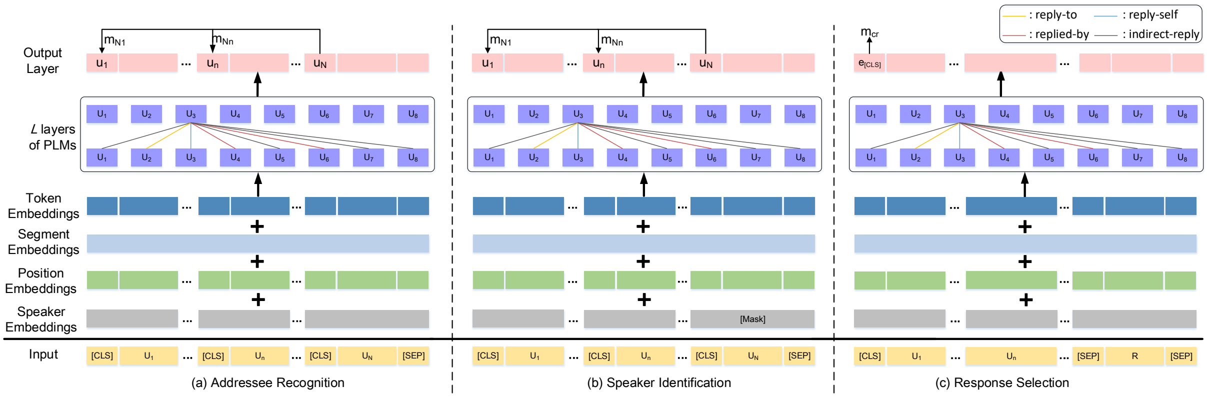


Figure 2: Input representations and model architectures when fine-tuning on (a) addressee recognition, (b) speaker identification and (c) response selection. Specifically for U_3, it illustrates how the graph-induced signals of the conversation structure in Figure 1(b) are utilized during Transformer-based encoding.

the input representation as shown in Figure 2, to consider the speaker information of each utterance. Considering that the set of interlocutors are inconsistent in different conversations, a position-based interlocutor embedding table is initialized randomly at first and is updated during fine-tuning. In this way, each interlocutor in a conversation is assigned with an embedding vector according to the order it appears in the conversation. Then, the speaker embeddings for each utterance can be derived by looking up this embedding table and assigned for all tokens in this utterance. The speaker embeddings are combined with the standard token, position and segmentation embeddings. The input representation is denoted as $\mathbf{H} = \{\mathbf{h}_m\}_{m=0}^{M}$, where $\mathbf{h}_m \in \mathbb{R}^d$, d is the dimension of embedding vectors and M is the length of input sequences.

3.3 Graph-Induced Encoding

To derive the contextualized and graph-induced representations, the output of encoding of our proposed method is based on both *semantic similarity* and *structural relationships* between a query vector and each of a set of key vectors. Given the input representation $\mathbf{H}$, it is first encoded with the multi-head self-attention mechanism as

$$\text{head}_i = \text{Atten}(\mathbf{HW}_i^q, \mathbf{HW}_i^k, \mathbf{HW}_i^v), \quad (1)$$

$$\text{MultiHead}(\mathbf{H}) = [\text{head}_1, ..., \text{head}_h]\mathbf{W}^o, \quad (2)$$

where $\mathbf{W}_i^q \in \mathbb{R}^{d\times\frac{d}{h}}$, $\mathbf{W}_i^k \in \mathbb{R}^{d\times\frac{d}{h}}$, $\mathbf{W}_i^v \in \mathbb{R}^{d\times\frac{d}{h}}$ and $\mathbf{W}^o \in \mathbb{R}^{d\times d}$ are all trainable parameters. h is the number of attention heads and [;] denotes the concatenation operation.

When calculating attention weights between tokens, existing Transformer-based PLMs consider the relationship between any two tokens to be equivalent. This approach does not model the inherent graph structure while encoding, which is crucial for constructing a graph-induced topology. To distinguish different relationships between utterances, edge-type-dependent parameters $\phi(e_{q,v})$ are utilized to refine the attention weights as

$$\text{Atten}(q, k, v) = \text{softmax}(\phi(e_{q,v})\frac{\mathbf{q}^\top\mathbf{k}}{\sqrt{d}})\mathbf{v}, \quad (3)$$

where $e_{q,v} \in$ {*reply-to, replied-by, reply-self, indirect-reply*} as illustrated in Figure 1(b). On the one hand, the *reply-to* edge guides the modeling of what the current utterance should be like given the prior utterance it replies to. On the other hand, the *replied-by* edge focuses on how the posterior utterances amend the modeling of the current utterance. In addition, the *reply-self* edge determines how much of the original semantics should be kept. Finally, the rest of the utterances are connected through the *indirect-reply* edge for contextualization. It is notable that the relationships between utterances are assigned for all tokens in an utterance. With these four types of edges, different relationships between utterances can be distinguished and the contextualized encoding can be conducted following a graph-induced topology. The dependency of an utterance on another can be well modeled for better MPC understanding.

Afterwards, the operations of residual connection, layer normalization and feed-forward network are applied accordingly as those used in a standard Transformer encoder layer (Vaswani et al., 2017). Finally, the combination of all the above operations is performed L times to derive deep contextualized representations for MPC understanding.

4 Downstream Tasks

Three downstream tasks are employed to evaluate the MPC understanding as comprehensively as possible, aiming at the issues of addressing whom, who speaking and saying what. When fine-tuning on each downstream task, all parameters are updated. Figure 2 shows the input representations and model architectures for three tasks respectively.

4.1 Addressee Recognition

In this paper, we follow the experimental setting in Ouchi and Tsuboi (2016) and Zhang et al. (2018) where models are tasked to recognize the addressee of the last utterance in a conversation.[1] Formally, models are asked to predict $\hat{a}_N$ given $\{(s_n, u_n, a_n)\}_{n=1}^{N} \backslash a_N$, where $\hat{a}_N$ is selected from the interlocutor set in this conversation and $\backslash$ denotes exclusion. When fine-tuning, this task is reformulated as finding a preceding utterance from the same addressee.

U_n is a sequence of utterance tokens. A [CLS] token is inserted at the start of each utterance, denoting the utterance-level representation for each individual utterance. Then, all utterances in a conversation are concatenated and a [SEP] token is inserted at the end of the whole sequence. It is notable that the reply-to edge of the last utterance is masked to avoid leakage. After encoded by PLMs, the contextualized representations for each [CLS] token representing individual utterances are extracted. A task-dependent non-linear transformation layer is placed on top of PLMs in order to adapt the output of PLMs to different tasks. Next, a layer normalization is performed to derive the utterance representations for this specific task $\{\mathbf{u}_n\}_{n=1}^{N}$, where $\mathbf{u}_n \in \mathbb{R}^d$. Then, for the last utterance U_N, its reply-to matching scores with all its preceding utterances are calculated as

$$m_{Nn} = \mathrm{softmax}(\mathbf{u}_N^\top \cdot \mathbf{A} \cdot \mathbf{u}_n),\ n < N, \quad (4)$$

where m_{Nn} is defined as the probability of the speaker of U_n being the addressee of U_N. Then, the utterance with the highest score is selected and the speaker of the selected utterance is considered as the recognized addressee. Finally, the fine-tuning objective of this task is to minimize the cross-entropy loss as

$$\mathcal{L}_{ar} = -\sum_{n=1}^{N-1} y_{Nn}\, log(m_{Nn}), \quad (5)$$

where $y_{Nn} = 1$ if the speaker of U_n is the addressee of U_N and $y_{Nn} = 0$ otherwise.

4.2 Speaker Identification

We follow the experimental setting in Gu et al. (2021) where models are tasked to identify the speaker of the last utterance in a conversation. Formally, models are asked to predict $\hat{s}_N$ given $\{(s_n, u_n, a_n)\}_{n=1}^{N} \backslash s_N$, where $\hat{s}_N$ is selected from the interlocutor set in this conversation. When fine-tuning, this task is reformulated as identifying the utterances sharing the same speaker.

First, the speaker embedding of the last utterance in the input representation is masked to avoid information leakage. Similar to the task of addressee recognition, the operations of PLM encoding, extracting the representations for [CLS] tokens, non-linear transformation and layer normalization are performed. For the last utterance U_N, its identical-speaker matching scores m_{Nn} with all preceding utterances are calculated similarly as Eq. (4). Here, m_{Nn} denotes the probability of U_N and U_n sharing the same speaker. The fine-tuning objective of this task is to minimize the cross-entropy loss similarly as Eq. (5). Here, $y_{Nn} = 1$ if U_n shares the same speaker with U_N and $y_{Nn} = 0$ otherwise.

4.3 Response Selection

This task asks models to select $\hat{u}_N$ from a set of response candidates given the conversation context $\{(s_n, u_n, a_n)\}_{n=1}^{N} \backslash u_N$, which is an important retrieval-based approach for chatbots. The key is to measure the similarity between two segments of context and response.

Formally, utterances in a context are first concatenated to form a segment, and each response candidate is the other segment. Then, the two segments are concatenated with a [SEP] token and a [CLS] token is inserted at the beginning of the whole sequence.

The contextualized representation $\mathbf{e}_{\texttt{[CLS]}}$ for the first [CLS] token using PLMs is extracted, which is an aggregated representation containing the semantic matching information for the context-response pair. Then, $\mathbf{e}_{\texttt{[CLS]}}$ is fed into a non-linear transformation with sigmoid activation to obtain

[1]We did not evaluate all utterances in an MPC, since the reply information of history utterances are utilized as the graph-induced signals which may cause information leakage.

Datasets		Train	Valid	Test
Hu et al. (2019)		311,725	5,000	5,000
Ouchi and Tsuboi (2016)	Len-5	461,120	28,570	32,668
	Len-10	495,226	30,974	35,638
	Len-15	489,812	30,815	35,385

Table 1: Statistics of the two benchmarks evaluated in this paper.

the matching score between the context and the response as

$$m_{cr} = \text{sigmoid}(\mathbf{e}_{\texttt{[CLS]}}^{\top} \cdot \mathbf{w} + b), \tag{6}$$

where m_{cr} denotes the probability of semantic matching between the context and the response candidate, $\mathbf{w} \in \mathbb{R}^{d \times 1}$ and $b \in \mathbb{R}^1$ are parameters updated during fine-tuning. Finally, the fine-tuning objective of this task is to minimize the cross-entropy loss according to the true/false labels of responses in the training set as

$$\mathcal{L}_{rs} = -[y_{cr} log(m_{cr}) + (1 - y_{cr}) log(1 - m_{cr})], \tag{7}$$

where $y_{cr} = 1$ if the response r is a proper one for the context c; otherwise $y_{cr} = 0$.

5 Experiments

5.1 Datasets

We evaluated our proposed methods on two Ubuntu IRC benchmarks. One was released by Hu et al. (2019), in which both speaker and addressee labels was provided for each utterance. The other benchmark was released by Ouchi and Tsuboi (2016). Here, we adopted the version shared in Le et al. (2019) for fair comparison. The conversation sessions were separated into three categories according to the session length (Len-5, Len-10 and Len-15) following the splitting strategy of previous studies (Ouchi and Tsuboi, 2016; Zhang et al., 2018; Le et al., 2019; Gu et al., 2021). Table 1 presents the statistics of the two benchmarks evaluated in our experiments.

5.2 Baseline Models

We compared the proposed method with (1) non-pre-training-based models including Preceding (Le et al., 2019), SRNN, DRNN (Ouchi and Tsuboi, 2016), SHRNN (Serban et al., 2016) and SIRNN (Zhang et al., 2018), as well as (2) pre-training-based models including BERT (Devlin et al., 2019), SA-BERT (Gu et al., 2020), and MPC-BERT (Gu et al., 2021). Readers can refer to Appendix A for implementation details of the baseline models.

5.3 Implementation Details

The base version of various PLMs were adopted for all our experiments. `GELU` (Hendrycks and Gimpel, 2016) was employed as the activation for all non-linear transformations. The Adam method (Kingma and Ba, 2015) was employed for optimization. The learning rate was initialized as 0.00002 and the warmup proportion was set to 0.1. Some configurations were different according to the characteristics of these datasets. For Hu et al. (2019), the maximum utterance number was set to 7 and the maximum sequence length was set to 230. For the three experimental settings in Ouchi and Tsuboi (2016), the maximum utterance numbers were set to 5, 10 and 15 respectively, and the maximum sequence lengths were set to 120, 220 and 320 respectively. For Hu et al. (2019), the fine-tuning process was performed for 10 epochs for addressee recognition, 10 epochs for speaker identification, and 5 epochs for response selection. For Ouchi and Tsuboi (2016), the fine-tuning epochs were set to 5, 5 and 3 for these three tasks respectively. The batch sizes were set to 16 for Hu et al. (2019), and 40, 20, and 12 for the three experimental settings in Ouchi and Tsuboi (2016) respectively. The fine-tuning was performed using a GeForce RTX 2080 Ti GPU. The validation set was used to select the best model for testing. All codes were implemented in the TensorFlow framework (Abadi et al., 2016) and are published to help replicate our results. [2]

5.4 Metrics and Results

Addressee recognition We followed the metric of previous work (Ouchi and Tsuboi, 2016; Zhang et al., 2018; Le et al., 2019; Gu et al., 2021) by employing precision@1 (P@1) to evaluate the performance of utterance prediction.

Table 2 presents the results of addressee recognition. It shows that GIFT helps improve the performance of all three PLMs on all test sets. In detail, BERT fine-tuned with GIFT (BERT w/ GIFT) outperformed its counterpart, i.e., fine-tuning BERT without graph-induced signals, by margins of 2.92%, 2.73%, 5.75% and 5.08% on these test sets respectively in terms of P@1. In addition, GIFT improved the performance of SA-

[2] https://github.com/JasonForJoy/MPC-BERT

	Hu et al. (2019)	Ouchi and Tsuboi (2016)		
		Len-5	Len-10	Len-15
Preceding (Le et al., 2019)	-	55.73	55.63	55.62
SRNN (Ouchi and Tsuboi, 2016)	-	60.26	60.66	60.98
SHRNN (Serban et al., 2016)	-	62.24	64.86	65.89
DRNN (Ouchi and Tsuboi, 2016)	-	63.28	66.70	68.41
SIRNN (Zhang et al., 2018)	-	72.59	77.13	78.53.
BERT (Devlin et al., 2019)	82.88	80.22	75.32	74.03
SA-BERT (Gu et al., 2020)	86.98	81.99	78.27	76.84
MPC-BERT (Gu et al., 2021)	89.54	84.21	80.67	78.98
BERT w/ GIFT	85.80†	82.95†	81.07†	79.11†
SA-BERT w/ GIFT	88.30†	84.49†	82.53†	82.65†
MPC-BERT w/ GIFT	**90.18**	**85.85**†	**84.13**†	**83.61**†

Table 2: Evaluation results of addressee recognition on the test sets in terms of P@1. Results except ours are cited from Ouchi and Tsuboi (2016) and Zhang et al. (2018). Numbers marked with † denoted that the improvements after implementing GIFT were statistically significant (t-test with p-value < 0.05) comparing with the corresponding PLMs. Numbers in bold denoted that the results achieved the best performance.

	Hu et al. (2019)	Ouchi and Tsuboi (2016)		
		Len-5	Len-10	Len-15
BERT (Devlin et al., 2019)	71.81	62.24	53.17	51.58
SA-BERT (Gu et al., 2020)	75.88	64.96	57.62	54.28
MPC-BERT (Gu et al., 2021)	83.54	67.56	61.00	58.52
BERT w/ GIFT	85.52†	89.74†	82.31†	80.40†
SA-BERT w/ GIFT	88.02†	90.01†	82.76†	80.87†
MPC-BERT w/ GIFT	**90.50**†	**90.61**†	**84.12**†	**81.51**†

Table 3: Evaluation results of speaker identification on the test sets in terms of P@1. Results except ours are cited from Gu et al. (2021).

BERT by margins of 1.32%, 2.50%, 4.26% and 5.22%, and of MPC-BERT by margins of 0.64%, 1.64%, 3.46% and 4.63% on these test sets respectively. These results verified the effectiveness and generalization of the proposed fine-tuning method.

Speaker identification Similarly, P@1 was employed as the evaluation metric of speaker identification for comparing performance.

Table 3 presents the results of speaker identification. It also shows that GIFT helps improve the performance of all three PLMs on all test sets. In detail, GIFT improved the performance of BERT by margins of 13.71%, 27.50%, 29.14% and 28.82%, of SA-BERT by margins of 12.14%, 25.05%, 25.14% and 26.59%, as well as of MPC-BERT by margins of 6.96%, 23.05%, 23.12% and 22.99% in terms of P@1 on these test sets respectively. From these results, we can see that the proposed fine-tuning method are particularly useful for speaker identification.

Response selection The $R_n@k$ metrics adopted by previous studies (Ouchi and Tsuboi, 2016; Zhang et al., 2018; Gu et al., 2021) were used here. Each model was tasked with selecting k best-matched responses from n available candidates for the given conversation context, and we calculated the recall of the true positive replies among the k selected responses, denoted as $R_n@k$. Two settings were followed in which k was set to 1, and n was set to 2 or 10.

Table 4 presents the results of response selection. Specifically, GIFT improved the performance of BERT by margins of 2.48%, 2.12%, 2.71% and 2.34%, of SA-BERT by margins of 3.04%, 4.16%, 5.18% and 5.35%, as well as of MPC-BERT by margins of 1.76%, 0.88%, 2.15% and 2.44% in terms of $R_{10}@1$ on these test sets respectively. From these results, we can get inspired that the graph-induced signals introduced to construct conversation structures were crucial for deep context understanding to select an appropriate response.

5.5 Discussions

Ablations To further illustrate the effectiveness of each component of the graph-induced topol-

	Hu et al. (2019)		Ouchi and Tsuboi (2016)					
			Len-5		Len-10		Len-15	
	R_2@1	R_{10}@1	R_2@1	R_{10}@1	R_2@1	R_{10}@1	R_2@1	R_{10}@1
DRNN (Ouchi and Tsuboi, 2016)	-	-	76.07	33.62	78.16	36.14	78.64	36.93
SIRNN (Zhang et al., 2018)	-	-	78.14	36.45	80.34	39.20	80.91	40.83
BERT (Devlin et al., 2019)	92.48	73.42	85.52	53.95	86.93	57.41	87.19	58.92
SA-BERT (Gu et al., 2020)	92.98	75.16	86.53	55.24	87.98	59.27	88.34	60.42
MPC-BERT (Gu et al., 2021)	94.90	78.98	87.63	57.95	89.14	61.82	89.70	63.64
BERT w/ GIFT	93.22†	75.90†	86.59†	56.07†	88.02†	60.12†	88.57†	61.26†
SA-BERT w/ GIFT	94.26†	78.20†	**88.07**†	**59.40**†	**89.91**†	**64.45**†	90.45†	65.77†
MPC-BERT w/ GIFT	**95.04**	**80.74**†	87.97	58.83†	89.77†	63.97†	**90.62**†	**66.08**†

Table 4: Evaluation results of response selection on the test sets. Results except ours are cited from Ouchi and Tsuboi (2016), Zhang et al. (2018) and Gu et al. (2021).

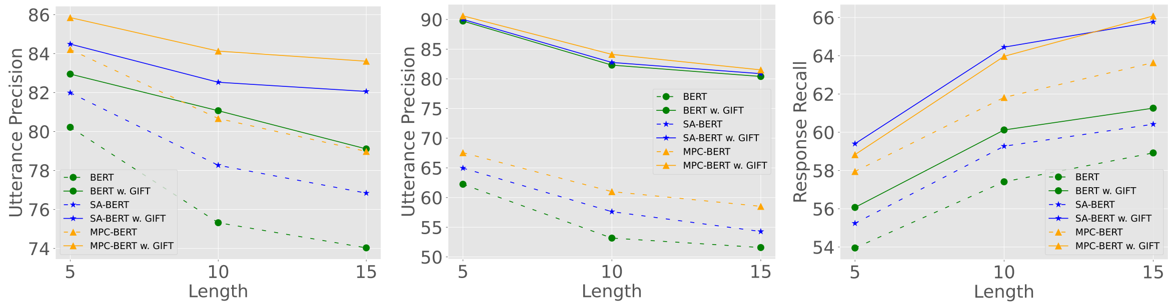
	AR (P@1)	SI (P@1)	RS (R_{10}@1)
BERT w/ GIFT	86.24	86.50	75.26
w/o reply-to and replied-by	84.38	70.67	72.30
w/o reply-to or replied-by	85.72	85.67	74.00
w/o reply-self	85.72	85.92	74.72
SA-BERT w/ GIFT	88.88	89.32	78.80
w/o reply-to and replied-by	86.90	77.07	77.50
w/o reply-to or replied-by	88.44	88.87	78.22
w/o reply-self	88.42	89.05	78.32
MPC-BERT w/ GIFT	90.78	91.72	81.08
w/o reply-to and replied-by	90.38	84.32	79.60
w/o reply-to or replied-by	90.52	90.90	80.22
w/o reply-self	90.46	91.10	80.02

Table 5: Evaluation results of the ablation tests on the validation set of Hu et al. (2019) on the tasks of addressee recognition (AR), speaker identification (SI), and response selection (RS).

ogy, three ablation tests were performed on the validation set of Hu et al. (2019) and the results were shown in Table 5. First, both reply-to and replied-by edges were ablated by merging these two types of edges with in-direct edges. The performance dropped significantly since these two types of edges constituted the majority of the conversation structure topology. Furthermore, reply-to or replied-by edges were ablated by merging these two types of edges together without distinguishing the bidirectional reply relationships between utterances. The performance drop verified the necessity of modeling what it uttered and what it received respectively. Finally, reply-self edges were merged with in-direct edges, showing that it is useful to distinguish self-replying from others.

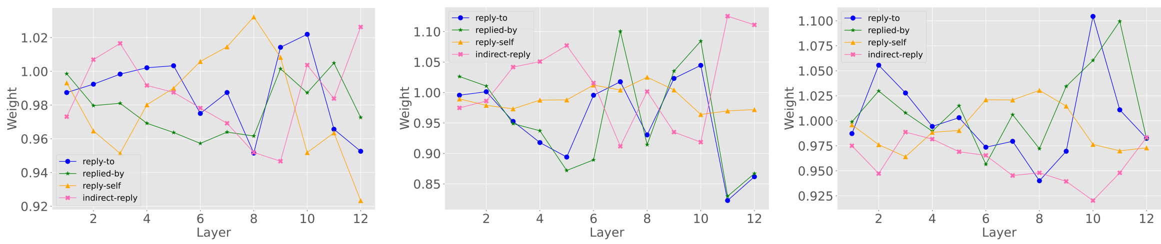
Impact of conversation length Figure 3 illustrated how the performance of BERT, SA-BERT and MPC-BERT, as well as those implemented with GIFT changed with respect to different session lengths on three downstream tasks and on the test sets of Ouchi and Tsuboi (2016). First, we can draw the conclusions that the performance of addressee recognition and speaker identification dropped, while the performance of response selection was significantly improved for all models as the session length increased, which was consistent with the findings in Gu et al. (2021). Furthermore, to quantitatively compare the performance difference at different session lengths, the performance margins between Len-5 and Len-10, as well as those between Len-10 and Len-15 were calculated. Readers can refer to Table 6 in Appendix B for details of these margins. From the results, it can be seen that as the session length increased, the performance of models with GIFT dropped more slightly on addressee recognition and speaker identification, and enlarged more on response selection, than the models without GIFT in most 14 out of 18 cases (including every 2 margins across lengths 5-10-15 for each model on each task). These results implied the superiority of introducing graph-induced signals on modeling long MPCs with complicated structures.

Visualization of weights Figure 4 visualized how the weights of four types of edges changed with respect to different encoding layers on three downstream tasks. Here, we took MPC-BERT fine-tuned on the training set of Hu et al. (2019) as an example. On the one hand, we can see that the changing trends of reply-to and replied-by edges were roughly the same, illustrating that these two types of edges were closely related to each other. Meanwhile, the values of these two edges were always different, further verifying the necessity of distinguishing the bidirectional reply relationships. On the other hand, the indirect-



(a) Addressee Recognition (b) Speaker Identification (c) Response Selection

Figure 3: Performance of models fine-tuned with or without graph-induced signals at different session lengths on the test sets of Ouchi and Tsuboi (2016) of three downstream tasks.



(a) Addressee Recognition (b) Speaker Identification (c) Response Selection

Figure 4: The weights of four types of edges in different encoding layers of MPC-BERT fine-tuned on the training set of Hu et al. (2019) of three downstream tasks.

reply edges generally followed the trend of first rising, then falling, and finally rising again. In addition, the values of this edge were always the minimum among all four edges at the beginning, and surprisingly became the maximum in the last layer (to clarify, 0.9834, 0.9825 and 0.9821 for indirect-reply, reply-to and replied-by edges of the 12th layer in Figure 4(c) respectively). It is likely that models have learned human behavior in MPCs, i.e., paying less attention to utterances that are not the most relevant to themselves at first glance. After comprehending the most relevant utterances, turn to indirectly related ones in context for fully understanding the entire conversation.

6 Conclusion

In this paper, we present graph-induced fine-tuning (GIFT), a plug-and-play and lightweight method that distinguishes the relationships between utterances for MPC understanding. The sparse but distinctive dependency of an utterance on another among those in an MPC is modeled by utilizing the edge-type-dependent parameters to refine the attention weights during the internal encoding of PLMs. Experimental results on three downstream tasks show that GIFT significantly helps improve the performance of three PLMs and achieves new state-of-the-art performance on two benchmarks. Obviously, the addressee labels of utterances in the conversation history are important for building the inherent graph structure required for graph-induced fine-tuning. However, an MPC with a few addressee labels missing is a common issue. In the future, it will be part of our work to investigate the scarcity of addressee labels.

Limitations

Enabling dialogue agents to join multi-party conversations naturally is undoubtedly a crucial step towards building human-like conversational AI, especially as such technology becomes more affordable and portable. More crucially, research on multi-party conversations has the promising potential to improve the interactive experience between humans and machines. Although the proposed method has shown great performance and generalization ability across various models and tasks, however, we never lose the sight of the other side of the coin. The proposed method requires full interactions among utterances in multihead attention of Transformers. Therefore, computational complexity and inference latency may be worth considering when deploying to online dialogue systems. Aside from the well-known difficulties in deployment, the proposed method was only evaluated on the domain-specific datasets, i.e., Ubuntu IRC, considering the constraints of dataset resources. In the future, we will try to search more open-domain datasets for multi-party conversations, and test if the proposed method can still show great performance on a more challenging open-domain setting.

Acknowledgements

This work was supported by the Opening Foundation of State Key Laboratory of Cognitive Intelligence, iFLYTEK COGOS-2022005. We thank anonymous reviewers for their valuable comments.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016.*, pages 265–283.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware BERT for multi-turn response selection in retrieval-based chatbots. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2041–2044.

Jia-Chen Gu, Chao-Hong Tan, Chongyang Tao, Zhen-Hua Ling, Huang Hu, Xiubo Geng, and Daxin Jiang. 2022. HeterMPC: A heterogeneous graph neural network for response generation in multi-party conversations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5086–5097. Association for Computational Linguistics.

Jia-Chen Gu, Chongyang Tao, Zhen-Hua Ling, Can Xu, Xiubo Geng, and Daxin Jiang. 2021. MPC-BERT: A pre-trained language model for multi-party conversation understanding. 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 3682–3692. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Wenpeng Hu, Zhangming Chan, Bing Liu, Dongyan Zhao, Jinwen Ma, and Rui Yan. 2019. GSN: A graph-structured network for multi-party dialogues. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5010–5016.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph Peper, Vignesh Athreya, R. Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C. Polymenakos, and Walter S. Lasecki. 2019. A large-scale corpus for conversation disentanglement. 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 3846–3856.

Ran Le, Wenpeng Hu, Mingyue Shang, Zhenjun You, Lidong Bing, Dongyan Zhao, and Rui Yan. 2019. Who is speaking to whom? learning to identify utterance addressee in multi-party conversations. 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 1909–1919.

Hui Liu, Zhan Shi, Jia-Chen Gu, Quan Liu, Si Wei, and Xiaodan Zhu. 2020. End-to-end transition-based online dialogue disentanglement. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3868–3874. ijcai.org.

Hui Liu, Zhan Shi, and Xiaodan Zhu. 2021. Unsupervised conversation disentanglement through co-training. 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 2345–2356. Association for Computational Linguistics.

Zhao Meng, Lili Mou, and Zhi Jin. 2018. Towards neural speaker modeling in multi-party conversation: The task, dataset, and models. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Hiroki Ouchi and Yuta Tsuboi. 2016. Addressee and response selection for multi-party conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2133–2143.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 300–325. Association for Computational Linguistics.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3776–3784.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1577–1586.

Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. One time of interaction may not be enough: Go deep with an interaction-over-interaction network for response selection in dialogues. 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 1–11.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Weishi Wang, Steven C. H. Hoi, and Shafiq R. Joty. 2020. Response selection for multi-party conversations with dynamic topic tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6581–6591.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 496–505.

Rui Zhang, Honglak Lee, Lazaros Polymenakos, and Dragomir R. Radev. 2018. Addressee and response selection in multi-party conversations with speaker interaction rnns. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5690–5697.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 270–278. Association for Computational Linguistics.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1118–1127.

A Baseline Models

We compared GIFT with these baseline methods.

A.1 Non-pre-training-based Models

- **Preceding** Le et al. (2019) was a heuristic method where the addressee was designated as the preceding speaker of the current speaker.

- **SRNN** and **DRNN** Ouchi and Tsuboi (2016) proposed the static or dynamic recurrent neural network-based models (SRNN or DRNN) where the speaker embeddings were fixed or updated with the conversation flow.

- **SHRNN** Inspired by Serban et al. (2016), Zhang et al. (2018) implemented Static-Hier-RNN (SHRNN), a hierarchical version of SRNN. It first built utterance embeddings from words and then processed utterance embeddings using high-level RNNs.

- **SIRNN** Zhang et al. (2018) proposed a speaker interaction RNN-based model (SIRNN). This model distinguished the interlocutor roles (sender, addressee, observer) at a finer granularity and updated the speaker embeddings role-sensitively, since interlocutors might play one of the three roles at each turn and those roles vary across turns.

A.2 Pre-training-based Models

The proposed GIFT was implemented into three PLMs.

- **BERT** (Devlin et al., 2019) was pre-trained to learn universal language representations on a large amount of general corpora with the self-supervised tasks of MLM and NSP.

- **SA-BERT** (Gu et al., 2020) added speaker embeddings and further pre-trained BERT on a domain-specific corpus to incorporate domain knowledge. We re-implemented SA-BERT on the same pre-training corpus used in this paper to ensure fair comparison.

- **MPC-BERT** (Gu et al., 2021) was pre-trained with two major types of self-supervised tasks for modeling interlocutor structures and utterance semantics in a unified framework.

	Len 5 → Len 10	Len 10 → Len 15
	AR (P@1)	
BERT	-4.90	-1.29
BERT w. GIFT	-1.88‡	-1.96
SA-BERT	-3.72	-1.43
SA-BERT w. GIFT	-1.96‡	-0.47‡
MPC-BERT	-3.54	-1.69
MPC-BERT w. GIFT	-1.72‡	-0.52‡
	SI (P@1)	
BERT	-9.07	-1.59
BERT w. GIFT	-7.43‡	-1.91
SA-BERT	-7.34	-3.34
SA-BERT w. GIFT	-7.25‡	-1.89‡
MPC-BERT	-6.56	-2.48
MPC-BERT w. GIFT	-6.49‡	-2.61
	RS (R_{10}@1)	
BERT	+3.46	+1.51
BERT w. GIFT	+4.05‡	+1.14
SA-BERT	+4.03	+1.15
SA-BERT w. GIFT	+5.05‡	+1.32‡
MPC-BERT	+3.87	+1.82
MPC-BERT w. GIFT	+5.14‡	+2.11‡

Table 6: Performance change of models as the session length increased on the test sets of Ouchi and Tsuboi (2016). For models with GIFT, numbers marked with ‡ denoted larger performance improvement or less performance drop compared with the corresponding models without GIFT.

B Impact of Conversation Length

To quantitatively compare the performance difference at different session lengths, the performance margins between Len-5 and Len-10, as well as those between Len-10 and Len-15 were calculated. Table 6 presents the details of these margins. From the results, it can be seen that as the session length increased, the performance of models with GIFT dropped more slightly on addressee recognition and speaker identification, and enlarged more on response selection, than the models without GIFT in most 14 out of 18 cases (including every 2 margins across lengths 5-10-15 for each model on each task).

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The section after conclusion.

☒ A2. Did you discuss any potential risks of your work?
The proposed method does not involve ethic concerns.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

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

B ☑ Did you use or create scientific artifacts?

Section 5

☑ B1. Did you cite the creators of artifacts you used?
Section 5

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 5

☒ 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)?
The employed pre-trained models and datasets are all open for academic research.

☒ 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?
Verification has been conducted before release by the original authors.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 5

☑ 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.
Appendix A.1

C ☑ Did you run computational experiments?

Section 5

☑ 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.3

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?
Appendix A.3

☑ 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?
Section 5

☒ 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.)?
No additional package was used.

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.