MPC-BERT: A Pre-Trained Language Model for Multi-Party Conversation Understanding

Jia-Chen Gu[1]*, **Chongyang Tao**[2], **Zhen-Hua Ling**[1], **Can Xu**[2], **Xiubo Geng**[2], **Daxin Jiang**[2]†
[1]National Engineering Laboratory for Speech and Language Information Processing,
University of Science and Technology of China, Hefei, China
[2]Microsoft, Beijing, China
gujc@mail.ustc.edu.cn, zhling@ustc.edu.cn,
{chotao,caxu,xigeng,djiang}@microsoft.com

Abstract

Recently, various neural models for multi-party conversation (MPC) have achieved impressive improvements on a variety of tasks such as addressee recognition, speaker identification and response prediction. However, these existing methods on MPC usually represent interlocutors and utterances individually and ignore the inherent complicated structure in MPC which may provide crucial interlocutor and utterance semantics and would enhance the conversation understanding process. To this end, we present MPC-BERT, a pre-trained model for MPC understanding that considers learning *who* says *what* to *whom* in a unified model with several elaborated self-supervised tasks. Particularly, these tasks can be generally categorized into (1) interlocutor structure modeling including reply-to utterance recognition, identical speaker searching and pointer consistency distinction, and (2) utterance semantics modeling including masked shared utterance restoration and shared node detection. We evaluate MPC-BERT on three downstream tasks including addressee recognition, speaker identification and response selection. Experimental results show that MPC-BERT outperforms previous methods by large margins and achieves new state-of-the-art performance on all three downstream tasks at two benchmarks.

1 Introduction

Speaker	Utterance	Addressee
I.1	How can I setup if I want add new server at xchat?	-
I.2	From places, network servers, work group, his computer, and then I clicked on the shared folder.	I.1
I.3	It did not allow you to see the files?	I.2
I.2	It prompts for authentication and I don't know what to put. I tried guest with no password.	I.3
I.4	Put proper authentication in, then?	I.2
I.3	I think you had kde on suse?	I.2

Table 1: An MPC example in Ubuntu IRC channel. Here, "I." is the abbreviation of "interlocutor".

Building a conversational agent with intelligence has drawn significant attention from both academia and industry. Most of existing methods have studied understanding conversations between two participants, aiming to return an appropriate response either in a generation-based (Shang et al., 2015; Serban et al., 2016, 2017; Zhang et al., 2018b, 2020) or retrieval-based manner (Lowe et al., 2015; Wu et al., 2017; Zhou et al., 2018; Tao et al., 2019a,b; Gu et al., 2019a,b, 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 conversation (MPC) (Ouchi and Tsuboi, 2016; Zhang et al., 2018a; Le et al., 2019; Hu et al., 2019). Table 1 shows an MPC example in the Ubuntu Internet Relay Chat (IRC) channel, which is composed of a sequence of (*speaker, utterance, addressee*) triples. In addition to returning an appropriate response, predicting who will be the next speaker (Meng et al., 2018) and who is the addressee of an utterance (Ouchi and Tsuboi, 2016; Zhang et al., 2018a; Le et al., 2019) are unique and important issues in MPC.

An instance of MPC always contains complicated interactions between interlocutors, between utterances and between an interlocutor and an utterance. Therefore, it is challenging to model the conversation flow and fully understand the dialogue content. Existing studies on MPC learn the representations of interlocutors and utterances with neural networks, and their representation

*Work done during the internship at Microsoft.

†Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing, pages 3682–3692
August 1–6, 2021. ©2021 Association for Computational Linguistics

spaces are either separate (Ouchi and Tsuboi, 2016) or interactive (Zhang et al., 2018a). However, the semantics contained in the interlocutor and utterance representations may not be effectively captured as they are from two different representation spaces. Recently, to take advantage of the breakthrough in pre-training language models (PLMs) for natural language understanding, some studies proposed to integrate the speaker (Gu et al., 2020) or topic (Wang et al., 2020) information into PLMs. Despite of the performance improvement on response selection, these models still overlook the inherent relationships between utterances and interlocutors, such as "address-to". Furthermore, most existing studies design models for each individual task in MPC (e.g., addressee recognition, speaker identification and response prediction) separately. Intuitively, these tasks are complementary among each other. Making use of these tasks simultaneously may produce better contextualized representations of interlocutors and utterances, and would enhance the conversation understanding, but is neglected in previous studies.

On account of above issues, we propose MPC-BERT which jointly learns *who* says *what* to *whom* in MPC by designing self-supervised tasks for PLMs, so as to improve the ability of PLMs on MPC understanding. Specifically, the five designed tasks includes *reply-to utterance recognition, identical speaker searching, pointer consistency distinction, masked shared utterance restoration* and *shared node detection*. The first three tasks are designed to model the interlocutor structure in MPC in a *semantics-to-structure* manner. In the output of MPC-BERT, an interlocutor is described through the encoded representations of the utterances it says. Thus, the representations of utterance semantics are utilized to construct the conversation structure in these three tasks. On the other hand, the last two tasks are designed to model the utterance semantics in a *structure-to-semantics* manner. Intuitively, the conversation structure influences the information flow in MPC. Thus, the structure information can also be used to strengthen the representations of utterance semantics in return. In general, these five self-supervised tasks are employed to jointly train the MPC-BERT in a multi-task learning framework, which helps the model to learn the complementary information among interlocutors and utterances, and that between structure and semantics. By this means, MPC-BERT can produce better interlocutor and utterance representations which can be effectively generalized to multiple downstream tasks of MPC.

To measure the effectiveness of these self-supervised tasks and to test the generalization ability of MPC-BERT, we evaluate it on three downstream tasks including *addressee recognition*, *speaker identification* and *response selection*, which are three core research issues of MPC. 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) and has three experimental settings according to session lengths. Experimental results show that MPC-BERT outperforms the current state-of-the-art models by margins of 3.51%, 2.86%, 3.28% and 5.36% on the test sets of these two benchmarks respectively in terms of the session accuracy of addressee recognition, by margins of 7.66%, 2.60%, 3.38% and 4.24% respectively in terms of the utterance precision of speaker identification, and by margins of 3.82%, 2.71%, 2.55% and 3.22% respectively in terms of the response recall of response selection.

In summary, our contributions in this paper are three-fold: (1) MPC-BERT, a PLM for MPC understanding, is proposed by designing five self-supervised tasks based on the interactions among utterances and interlocutors. (2) Three downstream tasks are employed to comprehensively evaluate the effectiveness of our designed self-supervised tasks and the generalization ability of MPC-BERT. (3) Our proposed MPC-BERT achieves new state-of-the-art performance on all three downstream tasks at two benchmarks.

2 Related Work

Existing methods on building dialogue systems can be generally categorized into studying two-party conversations and multi-party conversations (MPC). In this paper, we study MPC. In addition to predicting utterances, identifying the *speaker* and recognizing the *addressee* of an utterance are also important tasks for MPC. 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. (2018a) proposed SI-RNN, which updated speaker embeddings role-sensitively for addressee and response selection. Meng et al. (2018) proposed a task of speaker classification as a surrogate task for speaker modeling. Le et al.

(2019) proposed a who-to-whom (W2W) model to recognize the addressees of all utterances. Hu et al. (2019) proposed a graph-structured network (GSN) to model the graphical information flow for response generation. Wang et al. (2020) proposed to track the dynamic topic for response selection.

Generally speaking, previous studies on MPC cannot unify the representations of interlocutors and utterances effectively. Also, they are limited to each individual task, ignoring the complementary information among different tasks. To the best of our knowledge, this paper makes the first attempt to design various self-supervised tasks for building PLMs aiming at MPC understanding, and to evaluate the performance of PLMs on three downstream tasks as comprehensively as possible.

3 MPC-BERT and Self-Supervised Tasks

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 build a pre-trained language model for universal MPC understanding. Given a conversation, this model 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 downstream tasks by fine-tuning model parameters.

3.1 Model Overview

In this paper, BERT (Devlin et al., 2019) is chosen as the backbone of our PLM for MPC. Thus, we name it MPC-BERT. It is worth noting that our proposed self-supervised tasks for training MPC-BERT can also be applied to other types of PLMs.

We first give an overview of the input representations and the overall architectures of MPC-BERT. When constructing the input representations, in order to consider the speaker information of each utterance, *speaker* embeddings (Gu et al., 2020) are introduced as shown in Figure 1. Considering that the set of interlocutors are inconsistent in different conversations, a position-based interlocutor embedding table is initialized randomly at first and updated during pre-training, which means 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. The speaker embeddings are combined with standard token, position and segmentation embeddings and are then encoded by BERT. The output embeddings of BERT corresponding to different input tokens are utilized by different self-supervised tasks for further calculation.

3.2 Tasks of Interlocutor Structure Modeling

The first three tasks follow the *semantics-to-structure* manner. In MPC-BERT, each interlocutor is described through the encoded representations of the utterances it says. Thus, the representations of utterance semantics are utilized to construct the conversation structure. Figure 1 shows the input representations and the model architectures of these three tasks. A `[CLS]` token is inserted at the start of each utterance, denoting its utterance-level representation. 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 these three tasks share the same form of input data. Thus, the input only needs to be encoded once by BERT while the output can be fed into three tasks, which is computation-efficient. As shown in Figure 1, a task-dependent non-linear transformation layer is placed on top of BERT in order to adapt the output of BERT to different tasks. We will describe the details of these tasks as follows.

3.2.1 Reply-to Utterance Recognition

To enable the model to recognize the addressee of each utterance, a self-supervised task named *reply-to utterance recognition (RUR)* is proposed to learn which preceding utterance the current utterance replies to. After encoded by BERT, we extract the contextualized representations for each `[CLS]` token representing individual utterances. Next, a non-linear transformation followed by a layer normalization are performed to derive the utterance representations for this specific task $\{\mathbf{u}_i^{rur}\}_{i=1}^{N}$, where $\mathbf{u}_i^{rur} \in \mathbb{R}^d$ and $d = 768$. Then, for a specific utterance U_i, its matching scores with all its preceding utterances are calculated as

$$m_{ij} = \textbf{softmax}(\mathbf{u}_i^{rur\top} \cdot \mathbf{A}^{rur} \cdot \mathbf{u}_j^{rur}), \quad (1)$$

where $\mathbf{A}^{rur} \in \mathbb{R}^{d \times d}$ is a linear transformation, m_{ij} denotes the matching degree of U_j being the reply-to utterance of U_i, and $1 \le j < i$. We construct a set $\mathbb{S}$ by sampling a certain number of utterances

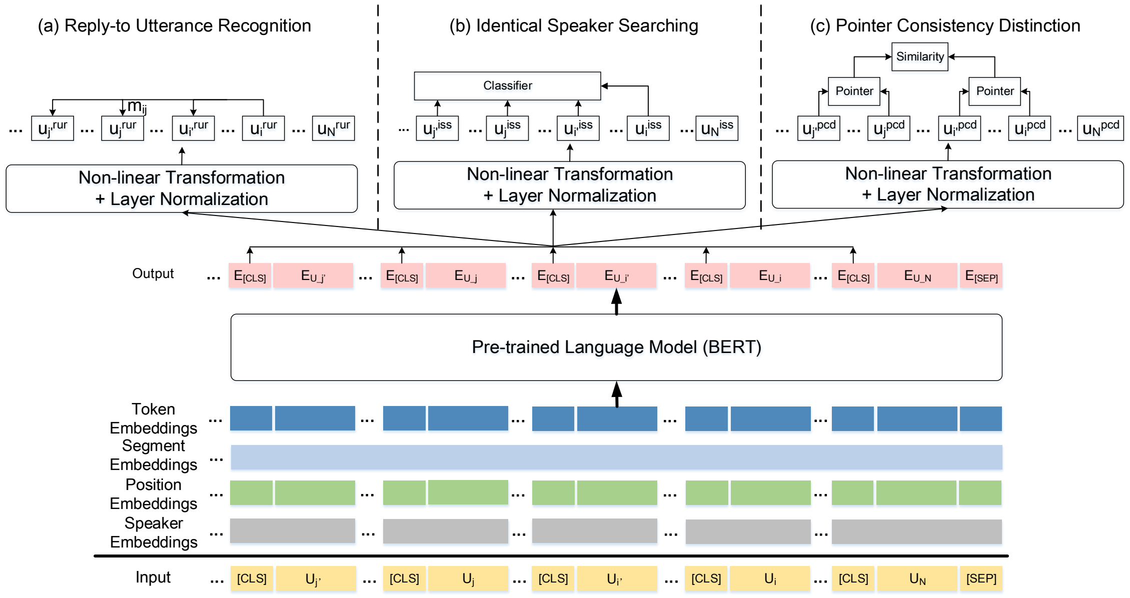

(a) Reply-to Utterance Recognition
(b) Identical Speaker Searching
(c) Pointer Consistency Distinction
m_ij
Classifier
Similarity
Pointer
Pointer
U_j'^rur U_j^rur U_i'^rur U_i^rur U_N^rur
U_j'^iss U_j^iss U_i'^iss U_i^iss U_N^iss
U_j'^pcd U_j^pcd U_i'^pcd U_i^pcd U_N^pcd
Non-linear Transformation + Layer Normalization
Output
E[CLS] EU_j' E[CLS] EU_j E[CLS] EU_i' E[CLS] EU_i E[CLS] EU_N E[SEP]
Pre-trained Language Model (BERT)
Token Embeddings
Segment Embeddings
Position Embeddings
Speaker Embeddings
Input
[CLS] Uj' [CLS] Uj [CLS] Ui' [CLS] Ui [CLS] UN [SEP]

Figure 1: Input representations and model architectures of the three self-supervised tasks for interlocutor structure modeling, including (a) reply-to utterance recognition, (b) identical speaker searching and (c) pointer consistency distinction.

in a conversation and this recognition operation is performed for each utterance in $\mathbb{S}$. Meanwhile, a dynamic sampling strategy is adopted so that models can see more samples. Finally, the pre-training objective of this self-supervised task is to minimize the cross-entropy loss as

$$\mathcal{L}_{rur} = -\sum_{i \in \mathbb{S}} \sum_{j=1}^{i-1} y_{ij} \, log(m_{ij}), \tag{2}$$

where $y_{ij} = 1$ if U$_j$ is the reply-to utterance of U$_i$ and $y_{ij} = 0$ otherwise.

3.2.2 Identical Speaker Searching

Having knowledge of who is the speaker of an utterance is also important for MPC. The task of *identical speaker searching (ISS)* is designed by masking the speaker embedding of a specific utterance in the input representation, and aims to predict its speaker given the conversation. Since the set of interlocutors vary across conversations, the task of predicting the speaker of an utterance is reformulated as *searching for the utterances sharing the identical speaker.*

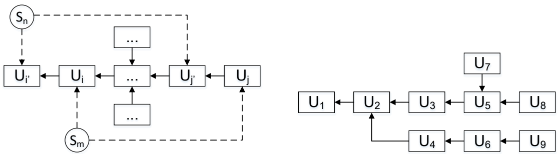
First, for a specific utterance, its speaker embedding is masked with a special `[Mask]` interlocutor embedding to avoid information leakage. Given the utterance representations for this specific task $\{\mathbf{u}_i^{iss}\}_{i=1}^N$ where $\mathbf{u}_i^{iss} \in \mathbb{R}^d$, the matching scores of U$_i$ with all its preceding utterances are calculated similarly with Eq. (1). Here, m_{ij} denotes the matching degree of U$_j$ sharing the same speaker with U$_i$. For each instance in the dynamic sampling set $\mathbb{S}$, there must be an utterance in previous turns sharing the same speaker. Otherwise, it is removed out of the set. Finally, the pre-training objective of this task is to minimize the cross-entropy loss similarly with Eq. (2). Here, $y_{ij} = 1$ if U$_j$ shares the same speaker with U$_i$ and $y_{ij} = 0$ otherwise.

3.2.3 Pointer Consistency Distinction

We design a task named *pointer consistency distinction (PCD)* to jointly model speakers and addressees in MPC. In this task, a pair of utterances representing the "*reply-to*" relationship is defined as a *speaker-to-addressee pointer*. Here, we assume that the representations of two pointers directing from the same speaker to the same addressee should be consistent. As illustrated in Figure 2 (a), speaker S$_m$ speaks U$_i$ and U$_j$ which reply to U$_{i'}$ and U$_{j'}$ from speaker S$_n$ respectively. Thus, the utterance tuples (U$_i$, U$_{i'}$) and (U$_j$, U$_{j'}$) both represent the pointer of S$_m$-to-S$_n$ and their pointer representations should be consistent..

Given the utterance representations for this specific task $\{\mathbf{u}_i^{pcd}\}_{i=1}^N$ where $\mathbf{u}_i^{pcd} \in \mathbb{R}^d$, we first capture the pointer information contained in each utterance tuple. The element-wise difference and multiplication between an utterance tuple (U$_i$, U$_{i'}$) are computed and are concatenated as

$$\mathbf{p}_{ii'} = [\mathbf{u}_i^{pcd} - \mathbf{u}_{i'}^{pcd}; \mathbf{u}_i^{pcd} \odot \mathbf{u}_{i'}^{pcd}], \tag{3}$$



(a) Pointer consistency distinction

(b) Shared node detection

Figure 2: Illustrations of the self-supervised tasks of (a) pointer consistency distinction and (b) shared node detection. Rectangles denote utterances, circles denote interlocutors, a solid line denotes an utterance replying to an utterance, and a dashed line denotes an utterance from an interlocutor.

where $\mathbf{p}_{ii'} \in \mathbb{R}^{2d}$. Then, we compress $\mathbf{p}_{ii'}$ and obtain the pointer representation $\bar{\mathbf{p}}_{ii'}$ as

$$\bar{\mathbf{p}}_{ii'} = \textbf{ReLU}(\mathbf{p}_{ii'} \cdot \mathbf{W}_{pcd} + \mathbf{b}_{pcd}), \tag{4}$$

where $\mathbf{W}_{pcd} \in \mathbb{R}^{2d \times d}$ and $\mathbf{b}_{pcd} \in \mathbb{R}^{d}$ are parameters. Identically, a consistent pointer representations $\bar{\mathbf{p}}_{jj'}$ and an inconsistent one $\bar{\mathbf{p}}_{kk'}$ sampled from this conversation are obtained. The similarities between every two pointers are calculated as

$$m_{ij} = \textbf{sigmoid}(\bar{\mathbf{p}}_{ii'}^{\top} \cdot \mathbf{A}^{pcd} \cdot \bar{\mathbf{p}}_{jj'}), \tag{5}$$

where m_{ij} denotes the matching degree of pointer $\bar{\mathbf{p}}_{ii'}$ being consistent with pointer $\bar{\mathbf{p}}_{jj'}$. m_{ik} can be derived accordingly. Finally, the pre-training objective of this task is to minimize the hinge loss which enforces m_{ij} to be larger than m_{ik} by at least a margin Δ as

$$\mathcal{L}_{pcd} = \textbf{max}\{0, \Delta - m_{ij} + m_{ik}\}. \tag{6}$$

3.3 Tasks of Utterance Semantics Modeling

Intuitively, the conversation structure might influence the information flow, so that it can be used to strengthen the representations of utterance semantics. Thus, two self-supervised tasks following the *structure-to-semantics* manner are designed.

3.3.1 Masked Shared Utterance Restoration

There are usually several utterances replying-to a shared utterance in MPC. Intuitively, a shared utterance is semantically relevant to more utterances in the context than non-shared ones. Based on this characteristic, we design a task named *masked shared utterance restoration (MSUR)*. We first randomly sample an utterance from all shared utterances in a conversation and all tokens in this sampled utterance are masked with a `[MASK]` token. Then the model is enforced to restore the masked utterance given the rest conversation.

Formally, assuming U$_i$ as the masked shared utterance and l_i as the number of tokens in U$_i$. Given the token representations for this task $\{\mathbf{u}_{i,t}^{msur}\}_{t=1}^{l_i}$ where $\mathbf{u}_{i,t}^{msur} \in \mathbb{R}^{d}$, the probability distribution of each masked token can be calculated as

$$\mathbf{p}_{u_{i,t}} = \textbf{softmax}(\mathbf{u}_{i,t}^{msur} \cdot \mathbf{W}_{msur} + \mathbf{b}_{msur}), \tag{7}$$

where $\mathbf{W}_{msur} \in \mathbb{R}^{d \times V}$ is the token embedding table, V denotes the vocabulary size, and $\mathbf{b}_{msur} \in \mathbb{R}^{V}$ is a bias vector. Finally, the pre-training objective of this self-supervised task is to minimize the negative log-likelihood loss as

$$\mathcal{L}_{msur} = -\frac{1}{l_i} \sum_{t=1}^{l_i} log\, p_{u_{i,t}}, \tag{8}$$

where $p_{u_{i,t}}$ is the element in $\mathbf{p}_{u_{i,t}}$ corresponding to the original token.

3.3.2 Shared Node Detection

A full MPC instance can be divided into several sub-conversations and we assume that the representations of sub-conversations under the same parent node tend to be similar. As illustrated in Figure 2 (b), two sub-conversations {U$_3$, U$_5$, U$_7$, U$_8$} and {U$_4$, U$_6$, U$_9$} share the same parent node U$_2$. Thus, they should be semantically relevant. Under this assumption, we design a self-supervised task named *shared node detection (SND)*, which utilizes the conversation structure to strengthen the capability of models on measuring the semantic relevance of two sub-conversations.

We first construct the pre-training samples for this task. Empirically, only the sub-conversations under the top shared node in a conversation are collected in order to filter out the sub-conversations with few utterances. Given a full MPC, the two sub-conversations with the most utterances form a positive pair. For each positive pair, we replace one of its elements with another sub-conversation randomly sampled from the training corpus to form a negative pair.

Formally, given two sub-conversations c_i and c_j, utterances in each sub-conversation are first concatenated respectively to form two segments. Then, the two segments are concatenated with a `[SEP]` token and a `[CLS]` token is inserted at the beginning of the whole sequence. This sequence are encoded by BERT to derive the contextualized representation for the [CLS] token. A non-linear transformation with sigmoid activation is further applied to this representation for calculating the matching score m_{ij}, i.e., the probability of c_i and c_j sharing the same parent node. Finally, the pre-training objective of this task is to minimize the cross-entropy loss as

$$\mathcal{L}_{snd} = -[y_{ij}log(m_{ij}) + (1 - y_{ij})log(1 - m_{ij})], \tag{9}$$

where $y_{ij} = 1$ if c_i and c_j share the same parent node and $y_{ij} = 0$ otherwise.

3.4 Multi-task Learning

In addition, we also adopt the tasks of masked language model (MLM) and next sentence prediction (NSP) in original BERT pre-training (Devlin et al., 2019), which have been proven effective for incorporating domain knowledge (Gu et al., 2020; Gururangan et al., 2020). Finally, MPC-BERT is trained by performing multi-task learning that minimizes the sum of all loss functions as

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{rur} + \mathcal{L}_{iss} + \mathcal{L}_{pcd} + \mathcal{L}_{msur} \\ + \mathcal{L}_{snd} + \mathcal{L}_{mlm} + \mathcal{L}_{nsp}.\end{aligned} \tag{10}$$

4 Downstream Tasks

4.1 Addressee Recognition

Given a multi-party conversation where part of the addressees are unknown, Ouchi and Tsuboi (2016) and Zhang et al. (2018a) recognized an addressee of the last utterance. Le et al. (2019) recognized addressees of all utterances in a conversation. In this paper, we follow the more challenging setting in Le et al. (2019).

Formally, models are asked to predict $\{\hat{a}_n\}_{n=1}^N$ given $\{(s_n, u_n, a_n)\}_{n=1}^N \backslash \{a_n\}_{n=1}^N$, where $\hat{a}_n$ is selected from the interlocutor set in this conversation and $\backslash$ denotes exclusion. When applying MPC-BERT, this task is reformulated as finding a preceding utterance from the same addressee. Its RUR matching scores with all preceding utterances are calculated following Eq. (1). 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

$$\mathcal{L}_{ar} = -\sum_{i=2}^{N}\sum_{j=1}^{i-1} y_{ij}\, log(m_{ij}), \tag{11}$$

where m_{ij} is defined in Eq. (1), $y_{ij} = 1$ if the speaker of U$_j$ is the addressee of U$_i$ and $y_{ij} = 0$ otherwise.

4.2 Speaker Identification

This task aims 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 applying MPC-BERT, this task is reformulated as identifying the utterances sharing the same speaker. For the last utterance U$_N$, its speaker embedding is masked and its ISS matching scores m_{Nj} with all preceding utterances are calculated following Section 3.2.2. The fine-tuning objective of this task is to minimize the cross-entropy loss as

$$\mathcal{L}_{si} = -\sum_{j=1}^{N-1} y_{Nj}\, log(m_{Nj}), \tag{12}$$

where $y_{Nj} = 1$ if U$_j$ shares the same speaker with U$_N$ and $y_{Nj} = 0$ otherwise.

4.3 Response Selection

This task asks models to select $\hat{u}_N$ from a set of response candidates given the conversation context $\{(s_n, u_n, a_n)\}_{n=1}^N \backslash u_N$. The key is to measure the similarity between two segments of context and response. We concatenate each response candidate with the context and extract the contextualized representation $\mathbf{e}_{\texttt{[CLS]}}$ for the first [CLS] token using MPC-BERT. Then, $\mathbf{e}_{\texttt{[CLS]}}$ is fed into a non-linear transformation with sigmoid activation to obtain the matching score between the context and the response. 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

$$\mathcal{L}_{rs} = -[ylog(m_{cr}) + (1-y)log(1-m_{cr})], \tag{13}$$

where $y = 1$ if the response r is a proper one for the context c; otherwise $y = 0$.

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

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 2: Statistics of the two benchmarks evaluated in this paper.

(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., 2018a; Le et al., 2019). Table 2 presents the statistics of the two benchmarks evaluated in our experiments.

5.2 Baseline Models

Non-pre-training-based models Ouchi and Tsuboi (2016) proposed a dynamic model DRNN which updated speaker embeddings with the conversation flow. Zhang et al. (2018a) improved DRNN to SI-RNN which updated speaker embeddings role-sensitively. Le et al. (2019) proposed W2W which jointly modeled interlocutors and utterances in a uniform framework, and predicted all addressees.

Pre-training-based models BERT (Devlin et al., 2019) was pre-trained to learn general language representations with MLM and NSP tasks. 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 with the pre-training corpus used in this paper to ensure fair comparison.

5.3 Implementation Details

The version of BERT-base-uncased was adopted for all our experiments. For pre-training, `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 0.00005 and the warmup proportion was set to 0.1. We pre-trained BERT for 10 epochs. The training set of the dateset used in Hu et al. (2019) was employed for pre-training. The maximum utterance number was set to 7. The maximum sequence length was set to 230. The maximum sampling numbers for each example were set to 4 for RUR, 2 for ISS and 2 for PCD. Δ in Eq. (6) was set to 0.4, achieving the best performance out of $\{0.2, 0.4, 0.6, 0.8\}$ on the validation set. The pre-training was performed using a GeForce RTX 2080 Ti GPU and the batch size was set to 4.

For fine-tuning, 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, and the maximum sequence lengths were set to 120, 220 and 320. All parameters in PLMs were updated. The learning rate was initialized as 0.00002 and the warmup proportion was set to 0.1. 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 respectively. The fine-tuning was also performed using a GeForce RTX 2080 Ti GPU. 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 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. [1]

5.4 Metrics and Results

Addressee recognition We followed the metrics of previous work (Le et al., 2019) by employing precision@1 (P@1) to evaluate each utterance with ground truth. Also, a session is marked as positive if the addressees of all its utterances are correctly recognized, which is calculated as accuracy (Acc.).

Table 3 presents the results of addressee recognition. It shows that MPC-BERT outperforms the best performing model, i.e., SA-BERT, by margins of 3.51%, 2.86%, 3.28% and 5.36% on these test sets respectively in terms of Acc., verifying the effectiveness of the proposed five self-supervised tasks as a whole. To further illustrate the effectiveness of each task, ablation tests were performed as shown in the last five rows of Table 3. We can observe that all self-supervised tasks are useful as removing any of them causes performance

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

	Hu et al. (2019)		Ouchi and Tsuboi (2016)					
			Len-5		Len-10		Len-15	
	P@1	Acc.	P@1	Acc.	P@1	Acc.	P@1	Acc.
Preceding (Le et al., 2019)	-	-	63.50	40.46	56.84	21.06	54.97	13.08
Subsequent (Le et al., 2019)	-	-	61.03	40.25	54.57	20.26	53.07	12.79
DRNN (Ouchi and Tsuboi, 2016)	-	-	72.75	58.18	65.58	34.47	62.60	22.58
SIRNN (Zhang et al., 2018a)	-	-	75.98	62.06	70.88	40.66	68.13	28.05
W2W (Le et al., 2019)	-	-	77.55	63.81	73.52	44.14	73.42	34.23
BERT (Devlin et al., 2019)	96.16	83.50	85.95	75.99	83.41	58.22	81.09	44.94
SA-BERT (Gu et al., 2020)	97.12	88.91	86.81	77.45	84.46	60.30	82.84	47.23
MPC-BERT	**98.31**	**92.42**	**88.73**	**80.31**	**86.23**	**63.58**	**85.55**	**52.59**
MPC-BERT w/o. RUR	97.75	89.98	87.51	78.42	85.63	62.26	84.78	50.83
MPC-BERT w/o. ISS	98.20	91.96	88.67	80.25	86.14	63.40	85.02	51.12
MPC-BERT w/o. PCD	98.20	91.90	88.51	80.06	85.92	62.84	85.21	51.17
MPC-BERT w/o. MSUR	98.08	91.32	88.70	80.26	86.21	63.46	85.28	51.23
MPC-BERT w/o. SND	98.25	92.18	88.68	80.25	86.14	63.41	85.29	51.39

Table 3: Evaluation results of addressee recognition on the test sets. Results except ours are cited from Le et al. (2019). Numbers in bold denote that the improvement over the best performing baseline is statistically significant (t-test with p-value < 0.05).

	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	**83.54**	**67.56**	**61.00**	**58.52**
MPC-BERT w/o. RUR	82.48	66.88	60.12	57.33
MPC-BERT w/o. ISS	77.95	66.77	60.03	56.73
MPC-BERT w/o. PCD	83.39	67.12	60.62	58.00
MPC-BERT w/o. MSUR	83.51	67.21	60.76	58.03
MPC-BERT w/o. SND	83.47	67.04	60.44	58.12

Table 4: Evaluation results of speaker identification on the test sets in terms of P@1. Numbers in bold denote that the improvement over the best performing baseline is statistically significant (t-test with p-value < 0.05).

drop. Among the five tasks, RUR plays the most important role, and the tasks focusing on modeling interlocutor structure contribute more than those for utterance semantics.

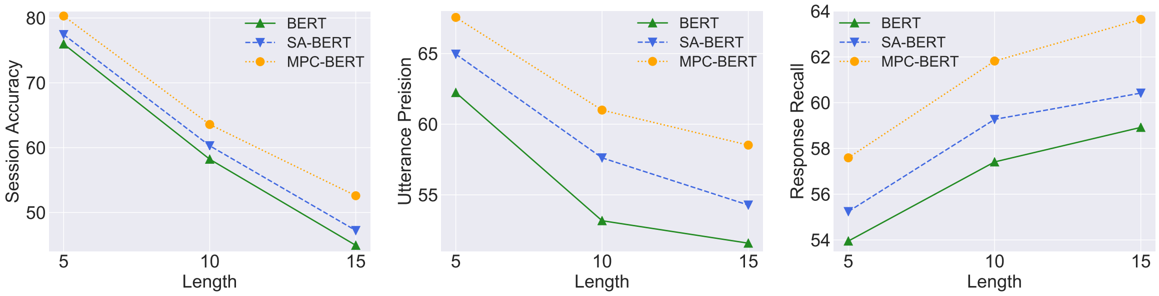
Speaker identification Similarly, P@1 was employed as the evaluation metric of speaker identification for the last utterance of a conversation and the results are shown in Table 4. It shows that MPC-BERT outperforms SA-BERT by margins of 7.66%, 2.60%, 3.38% and 4.24% respectively in terms of P@1. Besides, from the ablation results we find that all tasks are useful for improving the performance of speaker identification and ISS and RUR contribute the most. In particular, removing PCD, MSUR and SND only leads to slight performance drop. The reason might be that the information conveyed by these tasks is redundant.

Response selection The $\mathrm{R}_n@k$ metrics adopted by previous studies (Ouchi and Tsuboi, 2016; Zhang et al., 2018a) were used here. Each model was tasked with selecting k best-matched responses from n available candidates, and we calculated the recall as $\mathrm{R}_n@k$. Two settings were followed in which k was set to 1 and n was set to 2 or 10.

Table 5 presents the results of response selection. It shows that MPC-BERT outperforms SA-BERT by margins of 3.82%, 2.71%, 2.55% and 3.22% respectively in terms of $\mathrm{R}_{10}@1$. Ablation tests show that SND is the most useful task for response selection and the two tasks focusing on the utterance semantics contribute more than those

	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., 2018a)	-	-	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	**94.90**	**78.98**	**87.63**	**57.95**	**89.14**	**61.82**	**89.70**	**63.64**
MPC-BERT w/o. RUR	94.48	78.16	87.20	57.56	88.96	61.47	89.07	63.24
MPC-BERT w/o. ISS	94.58	78.82	87.54	57.77	88.98	61.76	89.58	63.51
MPC-BERT w/o. PCD	94.66	78.70	87.50	57.51	88.75	61.62	89.45	63.46
MPC-BERT w/o. MSUR	94.36	78.22	87.11	57.58	88.59	61.05	89.25	63.20
MPC-BERT w/o. SND	93.92	76.96	87.30	57.54	88.77	61.54	89.27	63.34

Table 5: Evaluation results of response selection on the test sets. Results except ours are cited from Ouchi and Tsuboi (2016) and Zhang et al. (2018a). Numbers in bold denote that the improvement over the best performing baseline is statistically significant (t-test with p-value < 0.05).



(a) Addressee recognition

(b) Speaker identification

(c) Response selection

Figure 3: Performance of models under different session lengths on the test sets of Ouchi and Tsuboi (2016) on the tasks of (a) addressee recognition, (b) speaker identification and (c) response selection.

focusing on the interlocutor structures.

5.5 Discussions

Figure 3 illustrates how the performance of BERT, SA-BERT and MPC-BERT changed with respect to different session lengths on the test sets of Ouchi and Tsuboi (2016). It can be seen that the performance of addressee recognition and speaker identification dropped as the session length increased. The reason might be that longer sessions always contain more interlocutors which increase the difficulties of predicting interlocutors. Meanwhile, the performance of response selection was significantly improved as the session length increased. It can be attributed to that longer sessions enrich the representations of contexts with more details which benefit response selection. Furthermore, as the session length increased, the performance of MPC-BERT dropped more slightly than that of SA-BERT on addressee recognition and speaker identification, and the R_{10}@1 gap between MPC-BERT and SA-BERT on response selection enlarged from 2.71% to 3.22%. These results imply the superiority of MPC-BERT over SA-BERT on modeling long MPCs with complicated structures.

6 Conclusion

In this paper, we present MPC-BERT, a pre-trained language model with five self-supervised tasks for MPC understanding. These tasks jointly learn *who* says *what* to *whom* in MPCs. Experimental results on three downstream tasks show that MPC-BERT outperforms previous methods by large margins and achieves new state-of-the-art performance on two benchmarks.

Acknowledgments

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, Zhen-Hua Ling, and Quan Liu. 2019a. Interactive matching network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 2321–2324.

Jia-Chen Gu, Zhen-Hua Ling, Xiaodan Zhu, and Quan Liu. 2019b. Dually interactive matching network for personalized response selection in retrieval-based chatbots. 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 1845–1854. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8342–8360.

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

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*.

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.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, pages 285–294.

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.

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.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3295–3301. AAAI Press.

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. 2019a. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 267–275. ACM.

Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019b. 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.

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. 2018a. 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, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018b. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1815–1825.

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.