SS-MPC: A Sequence-Structured Multi-Party Conversation System

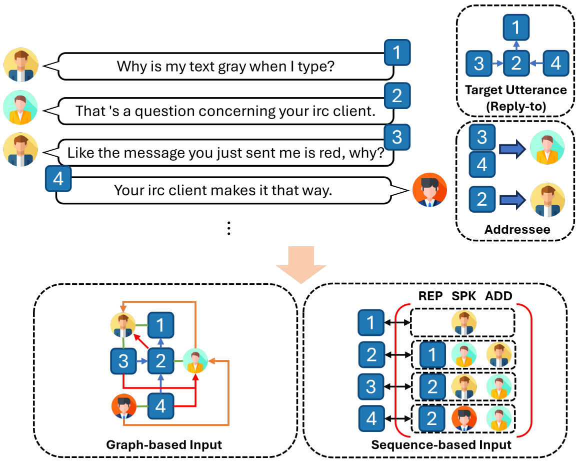
Anonymous ACL submission

Abstract

Recent Multi-Party Conversation (MPC) models typically rely on graph-based approaches to capture dialogue structures. However, these methods have limitations, such as information loss during the projection of utterances into structural embeddings and constraints in leveraging pre-trained language models directly. In this paper, we propose **SS-MPC**, a response generation model for MPC that eliminates the need for explicit graph structures. Unlike existing models that depend on graphs to analyze conversation structures, SS-MPC internally encodes the dialogue structure as a sequential input, enabling direct utilization of pre-trained language models. Experimental results show that **SS-MPC** achieves **15.60% BLEU-1** and **12.44% ROUGE-L** score, outperforming the current state-of-the-art MPC response generation model by **3.91%p** in **BLEU-1** and **0.62%p** in **ROUGE-L**. In addition, human evaluation confirms that SS-MPC generates more fluent and accurate responses compared to existing MPC models.[1]

1 Introduction

The rapid development of the Internet and the social media platforms have changed the way people communicate with each other and created new forms of interaction. In particular, Multi-Party Conversation (MPC), conversation in which multiple people freely exchanges opinions at the same time, is increasingly becoming common. MPC occurs on many platforms, such as group chats, online forums, and comment sections on social media. Recent research trends show that analysis and response generation on MPC is in its infancy, and the importance and need for them is increasingly being emphasized (Park and Lim, 2020; Anjum et al., 2020).



Why is my text gray when I type?

That 's a question concerning your irc client.

Like the message you just sent me is red, why?

Your irc client makes it that way.

Figure 1: An example of data in MPC Dataset (Ubuntu IRC Benchmark Dataset). The dataset is constructed of context and structural information. Context consists of utterances, and structural information consists of speaker information, target-utterance relation and addressee relation of each utterance.

MPC have flexibility to allow multiple speakers to participate in a conversation in no particular order or rule. These attributes of MPC complicate the flow and structure of the conversation compared to one-on-one conversations, and create additional challenges in understanding the context and intent of the utterances and speakers. Each speaker brings a unique context and intent to the conversation, and he or she must understand and process the context in which a particular utterance is being delivered to whom.

However, because the addressee of an utterance is often unclear in MPC, predicting or analyzing the structure of the dialogue is one of the major challenges for MPC. In addition, because multiple participants are simultaneously expressing their opinions and interacting, the topic and flow of the conversation are likely to change frequently. Unlike one-on-one conversations, these characteristics create the additional challenge of closely tracking

[1] The model investigated in this study is available at the following anonymous GitHub repository: `https://anonymous.4open.science/r/SS-MPC-51FD/README.md`

and managing the context of each utterance in MPC. For these reasons, developing systems for generating responses to MPC is considered one of the most challenging areas of current dialogue system research.

To address this complexity, MPC datasets typically include more structural conversation information for each utterance (Lowe et al., 2015).

For example, as shown in Figure 1, each utterance contains the structural information such as speaker information, which tells us who is speaking, the addressee information, which tells us to whom the utterance is addressed, and the target-utterance information, which indicates which utterance the current utterance is responding to.

The target-utterance relationship is typically linked to only one previous utterance in the conversation history, and the addressee is semantically the same as the speaker of the target-utterance. This structural information is critical for dialogue systems to understand and learn about the complexity of MPC, and it helps MPC response generation models generate responses that are appropriate for a given context.

However, traditional language models using sequential input have found that it is very difficult to express the structure of these conversations. Recent work (Gu et al., 2022, 2023b) has attempted to solve this problem using graphs. By representing utterances, speakers, and the relationships between utterances and speakers as graphs, and interpreting them through a graph encoder, it was possible to train a model to recognize the structure of a conversation and generate responses accordingly. But this still limits the use of pre-trained language models itself, since there are no such pre-trained graph-encoder models tuned properly to analyze the structure of conversations. If we partially employ randomly initialized graph-encoder, it may break the embedding space of the entire pre-trained model.

In this paper, we introduce **S**equence-**S**tructured **MPC** (SS-MPC), a response generation system for multi-party conversations that leverages the encoder-decoder architecture of the transformer while replacing the role of a graph encoder with **MPC structure tokens**. Instead of explicitly encoding conversational structures using a graph encoder, SS-MPC effectively integrates structural information through the well-designed soft prompts of the MPC structure tokens into the standard encoder-decoder framework. This approach eliminates the need for additional model components, accelerates the training process, and achieves superior performance compared to existing models.

Futhermore, unlike additinal models that require MPC analysis for the response generation, SS-MPC has the advantage of being able to analyze conversations and generate responses at once by end-to-end, enabling immediate usage in real-world MPC environments. By learning the conversation structure such as the correct target-utterance and addressee information for each utterance during the training process, the model can make appropriate inferences and generate the final response even when some of the correct target-utterance and addressee information is omitted during the actual inference process. This can be done because the SS-MPC contains the post-training process with the way masking the information partially, which means that the model itself is already trained for the ability to predict the omitted target-utterance or addressee information.

Our contributions are summarized as follows:

- We propose a novel method to train language models for MPC response generation without graph structures, which leverages sequence-structured inputs to internally represent the interaction flow in the dialogue.

- The proposed model can be used in real-world MPC environments easily because the model can simultaneously analyze conversations and generate responses using an end-to-end framework.

- Experimental results show that the proposed model performs better than the previous SOTA model. In addition, our various analysis results provide directions for future research in multi-party dialogues.

2 Related Work

2.1 Multi-Party Dialogue Structural Analysis

The task of predicting relationships between speakers to analyze the structure of MPC began in 2016. Ouchi and Tsuboi (2016) first proposed the addressee prediction and utterance selection task. To study this task, they first hand-created a dataset of MPC using log transcripts from Ubuntu IRC channels, and then utilized RNNs to perform the proposed task. Later, Zhang et al. (2018) proposed SI-RNN, which updates speaker embeddings based on roles for addressee prediction. Meng et al. (2017) also proposed a speaker classification task to model the relationships between speakers. Meanwhile, for the MPC response selection task, Wang et al. (2020) proposed to track dynamic topics, and then a who-to-whom (W2W) model (Le et al., 2019) was proposed to predict the addressees of all utterances in a conversation. Gu et al. (2021) proposed the MPC-BERT model, which utilizes multiple MPC learning methods to learn the complex interactions between recent utterances and interlocutors, and it performs post-training for MPC tasks. In addition, Gu et al. (2023a) proposed the GIFT model to help fine-tuning for MPC tasks with only simple scalar parameters on the attentions.

However, all the methodologies proposed in the above works have the limitation that they utilize the utterances to predict the addressee information of each utterance. This makes it difficult to utilize these methodologies for response generation models in real-world MPC environments.

2.2 Multi-Party Dialogue Response Generation

Along with these MPC tasks, there has been a parallel research on MPC response generation, which is the task of generating responses to a multi-party dialog. Hu et al. (2019) proposed a graph structure network (GSN) to model the graphical information flow for response generation. Later, Heter-MPC (Gu et al., 2022) was proposed to model complex interactions between utterances and interlocutors as graphs. This paper used graphs with two types of nodes and six types of edges to model the structure of multi-party conversations. Li and Zhao (2023) utilized the Expectation-Maximization (EM) algorithm in pre-training to predict the missing addressee information in the dataset. However, they still suffer from the drawback that the fine-tuning process only allows for an ideal setup where all addressees are labeled. To overcome this, MADnet (Gu et al., 2023b) utilizes the EM algorithm in the model of Gu et al. (2022) to directly predict and supplement the missing addressee information for training and response generation.

3 Methodology

In this paper, we propose SS-MPC, a novel MPC response generation model with the encoder-decoder structure of transformer. Here, we describe its input, output, and the training process.

3.1 Preliminaries

In a typical response generation task, the goal is to generate a final response $\bar{r}$ for a given conversation history h. In addition, the traditional MPC response generation task utilizes not only the dialogue history but also the dialogue structural information C. C consists of the speaker c_i, addressee a_i, and target-utterance u_i for each of the utterances.

$$C = \{c_1, c_2, \dots, c_n\} \tag{1}$$

$$c_i = \{s_i, a_i, u_i\} \tag{2}$$

where the number of the utterances is n and $1 \leq i \leq n$.

In general, the MPC datasets provide C, and the MPC models perform the task of finding the most appropriate response $\bar{r}$ based on the given h and C information. The response tokens are generated in an auto-regressive way. This can be formulated as follows:

$$\begin{aligned}\bar{r} &= \underset{r}{argmax}\ P(r|h, C; \theta) \\ &= \underset{r}{argmax} \prod_{t=1}^{|r|} P(r_t|h, C, r_{<t}; \theta)\end{aligned} \tag{3}$$

where r_t means the t-th token, and $r_{<t}$ means the previous tokens of the final response r.

The existing MPC models have utilized graph-based models to use C. In this process, there can be a loss of information when aligning the embedding space of h with C and using a graph encoder after random initialization.

$$\bar{r} = \underset{r}{argmax}\ P_{Dec}(r|GraphEnc(h, C); \theta) \tag{4}$$

where P_{Dec} means the probability of each token computed by the decoder of the model, and $GraphEnc(\cdot)$ means the embedding created by the graph encoder of the model.

In the case of SS-MPC, we use the sequence-structured input of each utterance, so we can utilize the full information of dialogue history and dialogue structural information without using a graph encoder.

$$\bar{r} = \underset{r}{argmax}\ P_{Dec}(r|Enc(h, s(C)); \theta) \tag{5}$$

where $s(\cdot)$ is the dialogue structuralization, which means transforming original input as the sequence-structured input containing the dialogue structure information internally.

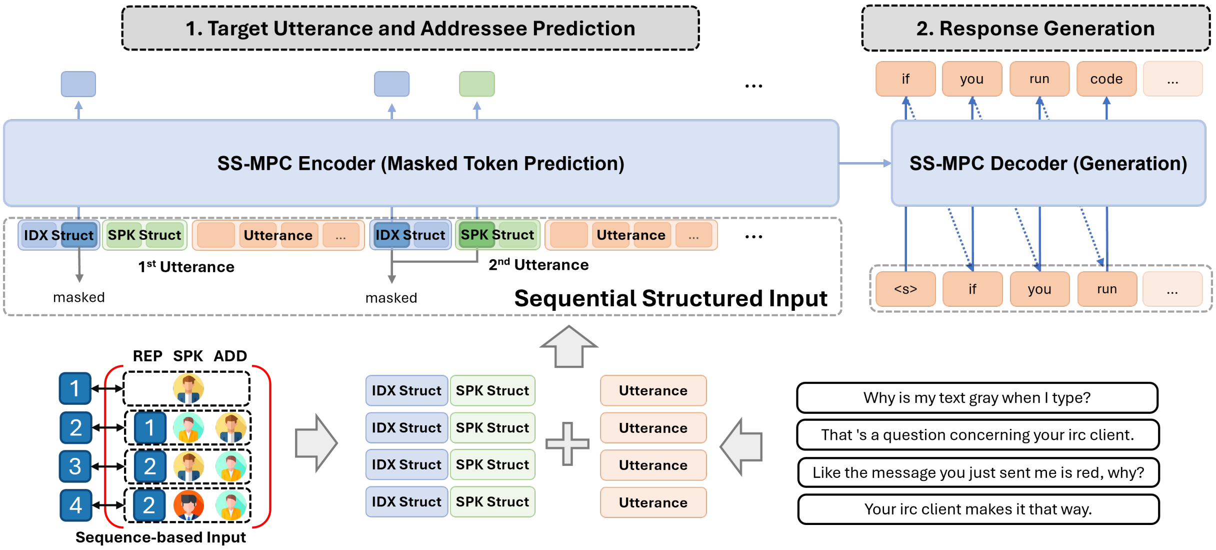

1. Target Utterance and Addressee Prediction
2. Response Generation
...
if
you
run
code
...
SS-MPC Encoder (Masked Token Prediction)
SS-MPC Decoder (Generation)
IDX Struct
SPK Struct
Utterance
...
IDX Struct
SPK Struct
Utterance
...
1st Utterance
2nd Utterance
masked
masked
Sequential Structured Input
<s>
if
you
run
...
REP SPK ADD
1
2
3
4
1
2
2
Sequence-based Input
IDX Struct
SPK Struct
Utterance
Why is my text gray when I type?
That 's a question concerning your irc client.
Like the message you just sent me is red, why?
Your irc client makes it that way.

Figure 2: The overview of the SS-MPC. The encoder part is expected to analyze the dialogue and predict the structural information in dialogue. The decoder part is expected to generate the final response with using the information analyzed in encoder.

Furthermore, we should consider the situation where the lack of structural information exists in MPC. Graph-encoder needs a fully connected graph since the lack of edges means that unconnected nodes can confuse the model in encoding. But SS-MPC does not need any change in model structure or additional training in this situation because the model is already trained with the way masking the information partially. The inference process of the SS-MPC is as follows:

$$\bar{r}, \bar{C} = \underset{r}{argmax}\ P(r|h, s(C_{omit}); \theta) \qquad (6)$$

where C_{omit} is the partially omitted structural information in MPC, and $\bar{C}$ means the predicted structural information for C_{omit}.

3.2 Overview of SS-MPC

Figure 2 shows the overview of the SS-MPC. It utilizes the encoder-decoder structure of the transformer because the transformer encoder is commonly used to analyze the structure of conversations (Shen et al., 2020; Mehri et al., 2019), and we want to leverage the strengths of these encoders to analyze the structure of conversations while allowing the decoder to focus on generating the actual responses.

The first step to utilize sequence-structured input instead of graph-structured input is to insert the conversation structure into a sequential form. To do this, we add three kinds of MPC structure tokens to the model tokenizer to reflect the structure of the conversation holding information about each utterance. The embeddings of new tokens act as soft prompts for each utterance, which reflect the structural information of the conversation.

Then we post-train only the encoder with the task of predicting dialogue structure to improve the encoder's ability to interpret the meaning of the added structure tokens and analyze the context with the dialogue structure. Through this process, the model not only learns the meaning of the added MPC structure tokens, but also learns to predict the partially omitted dialogue structure information of each utterance using the information of previous utterances and speakers.

3.3 MPC Structure tokens

To represent the speaker, addressee, and target-utterance for each utterance as a dialogue structure, three kinds of MPC structure tokens are added to the model tokenizer, called by MPC structure tokens; their embeddings are randomly initialized before training. The added MPC structure tokens are as follows:

- Index structure token
- Speaker structure token
- Structure masking token

Index structure token These tokens are developed to distinguish the order of the conversation; they are designed by indicating the order of the each utterance by number, such as "$[IDX_1]$",

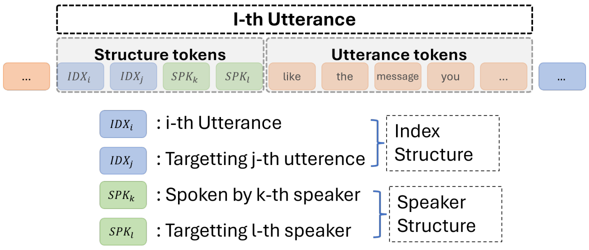


Figure 3: An example of the sequence-structure template for an utterance. Two index structure tokens which represents the utterance's index and the target-utterance's index, and two speaker structure tokens which represents speaker and addressee of the utterance are added as prefix to the tokenized utterance tokens.

"$[IDX_2]$", …,"$[IDX_n]$". The target-utterance can be also represented by specifying the index of the target-utterance.

Speaker structure token These tokens are to distinguish speakers in a conversation based on the order in which they appear; they are designed to distinguish between speakers, such as "$[SPK_1]$", "$[SPK_2]$", …, "$[SPK_m]$". The addressee information for each utterance can be expressed by specifying the speaker information via the corresponding token.

Structure masking token "$[Mask_{\text{IDX}}]$" and "$[Mask_{\text{SPK}}]$" tokens are added to mask MPC structure information. The "$[Mask_{\text{IDX}}]$" token masks the index and the target-utterance information, and the "$[Mask_{\text{SPK}}]$" token masks the speaker and the addressee information in a given utterance.

Since there is not a target-utterance and addressee for the first utterance, we add "$[IDX_{\text{None}}]$" token and "$[SPK_{\text{None}}]$" token to indicate that it has no target-utterance or addressee information.

3.4 Dialogue Structuralization

Dialogues are sequence-structured using MPC structure tokens that are added to express the structure of the conversation in a sequential form. The entire conversation can be broken down into utterances, and each of the utterances consists of the structure tokens and utterance tokens. Note that, for the final response, the utterance tokens are omitted because it is the answer which the model should generate.

In Figure 3, an example is shown for the i-th utterance of the entire conversation. The token that indicates that the i-th utterance is labeled by "$[IDX_i]$". This i-th utterance is responding to the j-th utterance, which labeled by the second token "$[IDX_j]$". Furthermore, the i-th utterance is being uttered by the speaker-k and is being answered to the speaker-l, which is represented by the tokens "$[SPK_k]$" and "$[SPK_l]$" in the following sequence. Thus we can set the sequence-structure template of i-th utterance as follows:

$$S_i = (\overbrace{[IDX_i];[IDX_j];[SPK_k];[SPK_l]}^{structure\ tokens}; \overbrace{[token_1];[token_2]...}^{utterance\ tokens}) \quad (7)$$

where $1 \leq i \leq n$ for dialogue with n utterances.

Then the model should generate the final response, so its structure information is inputted. The sequence-structure template is set as follows:

$$S_r = (\overbrace{[IDX_r];[IDX_t];[SPK_s];[SPK_a]}^{response\ structure\ tokens}) \quad (8)$$

where "$[IDX_r]$","$[IDX_t]$","$[SPK_s]$", and "$[SPK_a]$" tokens means the index, target-utterance index, speaker, and addressee of the current response utterance.

The sequence-structure templates of whole utterances in a dialogue are concatenated as a sequence-structured input $S = (S_1; S_2; ...; S_{n-1}; S_n; S_r)$. In addition, the structural information of the input can be masked using the "$[Mask_{\text{IDX}}]$" and "$[Mask_{\text{SPK}}]$" tokens. This masking approach is performed when target-utterance and addressee information has to be predicted in the conversation. In particular, the encoder is post-trained using the masking approach.

3.5 Post-training for Encoder

SS-MPC with the addition of MPC structure tokens needs to carry out post-training to obtain better embeddings of added tokens containing semantic and contextual information from context tokens. In particular, masking is performed with a probability of hyper-parameter $p\%$ for the structure tokens of each utterances, and the encoder predicts the correct answer for the masked structure tokens. To predict the structure token, the LM head of post-training for encoder shares the parameter with the LM head in decoder since they generate the same type of tokens; For the utterance tokens, we train the encoder to generate the tokens itself. The formulation of the loss function for post-training is as follows:

$$\mathcal{L}_{post} = -\sum_{i} \log P(x_i | X_{masked}; \phi) \qquad (9)$$

where i means each position of the input, x_i denotes the original encoder input token of the i-th position, for the masked encoder input X_{masked}. And the ϕ is the parameter of the SS-MPC encoder.

3.6 Fine-Tuning Model

SS-MPC is fine-tuned after post-training to perform the task of generating the final response. Fine-tuning is same as the learning process of a typical transformer encoder-decoder model. The difference is that the SS-MPC utilizes sequence-structured input.

$$\mathcal{L} = -\sum_{i=1}^{n} logP(r_i | r_{<i}, X; \theta) \qquad (10)$$

where r_i is the i-th final response token, X is the sequence-structured encoder input, and θ is the parameter of the SS-MPC.

4 Experiments

Dataset To evaluate the performance of the proposed SS-MPC model, we utilize the Ubuntu IRC benchmark dataset, which has originally released by Ouchi and Tsuboi (2016) and Hu et al. (2019), and has been widely using for various MPC tasks. This dataset comprises user conversations from the Internet Relay Chat (IRC) channel of the Ubuntu homepage.

Two Ubuntu IRC Benchmark datasets are used in the experiments as follows:

Ubuntu IRC (2016): The dataset released by Ouchi and Tsuboi (2016)[2] has some missing structural information in the dataset. We construct the sequence-structured input with masking those missing information. This dataset is categorized into three subsets based on session length (Len-5, Len-10, and Len-15). We employ the Len-5 subset, following the settings of preivious studies.

Ubuntu IRC (2019): The dataset released by Hu et al. (2019)[3] includes all structure information for every utterances. The dataset is used for post-training SS-MPC both in Ubuntu IRC (2016) and Ubuntu IRC (2019). Further details on the datasets can be found in the Appendix A.

Evaluation Metrics To evaluate SS-MPC, we just follow previous research and measure its performance using BLEU-1 through BLEU-4, METEOR, and ROUGE-L score for the final response. All metrics are computed using the Hugging Face `evaluate` library[4] (Wolf et al., 2020).

Baselines For the backbone model, we use **BART** (Lewis et al., 2020) as a widely recognized transformer-based encoder-decoder model. BART leverages the encoder-decoder architecture that are well-suited for response generation as well as other generative tasks such as summarization and machine translation.

We compare our approach against the following models: **(1) GSN** (Hu et al., 2019). GSN's core architecture consists of an utterance-level graph-structured encoder. **(2) GPT-2** (Radford et al., 2019), a unidirectional pre-trained language model. Following its original setup, all context utterances and response are concatenated by using a special token "$[SEP]$" as input. We also compare ConvMPC with the **(3) HeterMPC** (Gu et al., 2022) and **(4) MADnet** (Gu et al., 2023b), which are known as SOTA models among the current MPC response generation models. The HeterMPC model structures the speaker, target utterance, and addressee relationships in the form of a heterogeneous graph to model complex MPC. To analyze the structured data, it utilizes a heterogeneous graph encoder structure that utilizes Graph Attention (GAT) operations. In the case of the MADnet model, it is a model that slightly modifies the graph input of the existing HeterMPC model and adds the EM-algorithm methodology to generate a response by inferring the missing data from the existing data through the EM-algorithm.

We further evaluate our approach on decoder based Large Language Model (LLM) in appendix E.

Implementation Details Model parameters are initialized with **BART** (Lewis et al., 2020), both base and large, which were implemented in Hugging Face's `Transformers` library [5] (Wolf et al., 2020). We use AdamW (Loshchilov and Hutter, 2017) for optimization, with an initial learning rate of 5e-6 that decayed linearly. The model is trained

[2] We adopt the refined version provided by Le et al. (2019), which is released on https://github.com/lxchtan/HeterMPC (Gu et al., 2022)

[3] We adopt the re-emplemnted processed version of (Gu et al., 2022), which is released on https://github.com/lxchtan/HeterMPC

[4] https://huggingface.co/docs/evaluate/index

[5] https://huggingface.co/facebook/bart-large

Dataset	**Models**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**	**ROUGE-L**
Ubuntu IRC (2016)	GSN† (Hu et al., 2019)	10.23	3.57	1.70	0.97	4.10	9.91
	GPT-2 (Radford et al., 2019)	8.86	2.69	1.11	0.61	7.40	8.53
	BART (Lewis et al., 2020)	11.76	4.86	2.97	2.21	8.91	9.86
	MADNet (Gu et al., 2023b)	11.82	4.58	2.65	1.91	9.78	10.61
	SS-MPC$_{bart\_base}$	12.83	5.45	3.20	2.29	10.37	11.06
	SS-MPC$_{bart\_large}$	**13.27**	**5.66**	**3.38**	**2.44**	**10.77**	**11.37**
Ubuntu IRC (2019)	GSN† (Hu et al., 2019)	6.32	2.28	1.10	0.61	3.27	7.39
	GPT-2 (Radford et al., 2019)	10.85	3.76	1.61	0.84	9.00	7.24
	BART (Lewis et al., 2020)	12.71	4.52	2.13	1.25	8.75	10.01
	HeterMPC (Gu et al., 2022)	10.29	3.68	1.71	0.96	8.79	11.22
	MADNet (Gu et al., 2023b)	11.69	4.57	2.33	1.45	9.48	11.82
	SS-MPC$_{bart\_base}$	12.82	5.58	3.18	2.15	9.35	11.96
	SS-MPC$_{bart\_large}$	**15.60**	**6.62**	**3.67**	**2.44**	**10.92**	**12.44**

Table 1: Performance comparison of different models on two Ubuntu IRC Benchmark datasets (Ouchi and Tsuboi, 2016; Hu et al., 2019) with various metrics. The performance result of GSN† is cited from Gu et al. (2023b).

Human Evaluation	**Score**	**Kappa**
Gold Label	1.91	0.51
GPT-2 (Hu et al., 2019)	0.6	0.50
BART (Lewis et al., 2020)	1.50	0.48
MADNet (Gu et al., 2023b)	1.57	0.46
SS-MPC$_{bart\_large}$ **(Ours)**	**1.84**	0.55

Table 2: Human Evaluation results on Ubuntu IRC Benchmark test set of Ubuntu IRC (2019) Hu et al. (2019) with several MPC response generation models.

on the two Ubuntu IRC benchmark training sets, with a maximum of 10 epochs for post-training and fine-tuning individually. We use a batch size of 8 with 2 gradient accumulation steps and select the best model based on validation performance for testing. The maximum length of the generated output is set to 50 tokens just following previous studies.

4.1 Main Results

Table 1 shows model performances on two Ubuntu IRC Benchmark datasets, **Ubuntu IRC (2016)** (Ouchi and Tsuboi, 2016) and **Ubuntu IRC (2019)** (Hu et al., 2019). SS-MPC achieves a significant performance improvement on both datasets compared to previous models. On the **Ubuntu IRC (2016)** dataset, SS-MPC$_{bart\_base}$ outperforms the previous state-of-the-art (SOTA) model, MADNet, by 1.01%p in BLEU-1, 0.87%p in BLEU-2, 0.55%p in BLEU-3, 0.38%p in BLEU-4, 0.59%p in METEOR, and 0.45%p in ROUGE-L. Similarly, on the **Ubuntu IRC (2019)** dataset, SS-MPC$_{bart\_base}$ also surpasses MADNet by 1.13%p in BLEU-1, and 0.14%p in ROUGE-L.

It is important to note that previous SOTA models such as HeterMPC and MADNet partially replace components of the BART-base architecture with graph attention mechanisms. Therefore, a direct comparison with SS-MPC, which retains the original BART-base architecture, may be inappropriate. In fact, given the actual training and inference time under the same GPU usage, HeterMPC and MADNet are similar to SS-MPC$_{bart\_large}$ rather than SS-MPC$_{bart\_base}$.

You can see the response examples generated by each model in Appendix B and you can find the actual training time and inference time comparison with HeterMPC in Appendix C.

4.2 Human Evaluation

Since quantitative metrics alone may not fully capture the quality of generated responses, we also conduct a human evaluation. Specifically, we randomly sampled 100 conversations from the Ubuntu IRC benchmark dataset and asked three graduate students to evaluate the quality of the generated responses. The evaluation focuses on three independent aspects: (1) relevance, (2) fluency, and (3) informativeness. Each judge assigned binary scores for each aspect, with the final score ranging from 0 to 3. Table 2 presents the average final score of human evaluation results comparing GPT-2, BART, MADNet, and SS-MPC$_{bart\_large}$ against the ground truth (Gold Label). In addition, Fleiss's Kappa (Fleiss, 1971) is calculated to measure inter-annotator agreement. The result indicates that SS-MPC produces more relevant, fluent, and informative responses than any other model.

Further details on the instructions given to the judges for the human evaluation are provided in Appendix F

Models	Structural Info.	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
MADNet (Gu et al., 2023b)	fully given	11.69	4.57	2.33	1.45	9.48	11.82
SS-MPC$_{bart\_large}$		15.60	6.62	3.67	2.44	10.92	12.44
SS-MPC$_{real\text{-}world}$	-last utt.	13.59	5.47	3.10	2.14	9.15	11.03

Table 3: Performance of SS-MPC without structural information of last utterance.

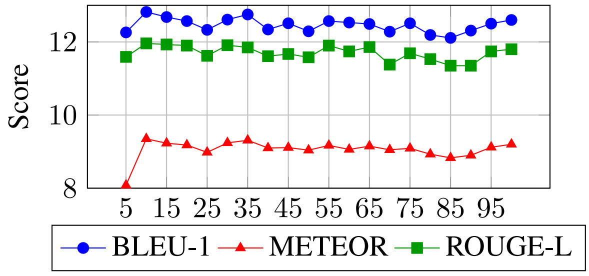


Figure 4: Performance of response generation with different masking probability on post-training

Model	Target Utt.	Adr.
BERT (Devlin et al., 2019)	-	82.88%
SA-BERT (Sun et al., 2019)	-	86.98%
MPC-BERT (Gu et al., 2021)	-	89.54%
GIFT (Gu et al., 2023a)	-	90.18%
SS-MPC$_{Encoder}$	76.38%	89.92%

Table 4: Performance (precision@1) of predicting target-utterance and addressee on the test set of Ubuntu IRC (2019) (Hu et al., 2019).

4.3 Ablation Study

Post-Training and Masking Probability Figure 4 shows how the performance of the model changes as the probability p of masking the target-utterance and addressee information varies during post-training.

As shown in Figure 4, we can see that the post-trained model with 10% of masking probability achieves the best performance for final response generation and models with masking probabilities above 10% obtain similar performances in response generation. This results provides the partial criteria of required masking probability when re-learning newly added token embeddings using the Masked Language Modeling (MLM) approach.

Addressee prediction The SS-MPC is trained to understand structures through structure tokens via post-training process. We hypothesize that this approach could be applied to the task of addressee prediction. To verify this, we train the model by masking only the addressee and predicting it. Table 4 shows the comparison of the addressee prediction tasks with other MPC analysis models, in terms of Precision@1. It shows that SS-MPC achieves 89.92% in addressee prediction, which is very close to the previous SOTA model, GIFT.

4.4 Response Generation in Real-World MPC Scenario

SS-MPC can generate responses even when partial structural information is missing by simply masking the absent tokens as Equation 6. Here, we assume the real-world MPC scenario, where the model should continuously generate the MPC. In this scenario, the target-utterance and addressee information of the response is missing. And the model has to predict the missing target-utterance and addressee while generating the following response. Our method can accumulate the predicted structure information to generate the next response continuously in this scenario. "$[IDX_t]$" and "$[SPK_a]$" are masked with structure masking tokens to construct sequence-structured input (Equation 8).

Table 3 presents the performance of SS-MPC finetuned without target-utterance and addressee information for the final response, which is marked as SS-MPC$_{real\text{-}world}$ in this table. While its performance is slightly lower than SS-MPC with full structural information, it is still comparable with the previous SOTA model, MADnet. This shows that SS-MPC can generate a response by predicting the target utterance and addressee simultaneously.

5 Conclusions

We introduce SS-MPC, a model optimized for generating responses in multi-party conversations. Unlike traditional graph-based approaches, SS-MPC employs the encoder-decoder architecture of transformer to fully leverage the pre-trained knowledge of language models. For this, we propose a novel method to encode dialogue structure sequentially within the input, allowing the model to capture the interaction flow in the dialogue without relying on explicit graph representations. SS-MPC outperforms the existing SOTA MPC response generation model and has the distinct advantage of easily application to real-world MPC scenarios depending on not requiring any additional module.

Limitations

The SS-MPC proposed in this paper has shown good performance compared to existing MPC response generation models, but it is limited by the fact that both training and inference are performed only on the Ubuntu IRC benchmark dataset, which makes it less generalizable. This is due to the absolute lack of MPC datasets, and it is necessary to apply the model to a wider variety of topics and conversations between different speakers to maintain generality. There is another room for further development of the model. For example, the model can be trained by initializing the initial embeddings of the added MPC structure tokens to specific values (e.g., [CLS] or [SEP] embeddings), or by initializing the embeddings to follow a specific distribution. Acquiring additional MPC datasets and further developing Multi-MPC will be part of our future work.

References

Omer Anjum, Chak Ho Chan, Tanitpong Lawphongpanich, Yucheng Liang, Tianyi Tang, Shuchen Zhang, Wen mei W. Hwu, Jinjun Xiong, and Sanjay J. Patel. 2020. Vertext: An end-to-end ai powered conversation management system for multi-party chat platforms. *Companion Publication of the 2020 Conference on Computer Supported Cooperative Work and Social Computing*.

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

Joseph Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–.

Jia-Chen Gu, Zhen-Hua Ling, QUAN LIU, Cong Liu, and Guoping Hu. 2023a. Gift: Graph-induced fine-tuning for multi-party conversation understanding. In *Annual Meeting of the Association for Computational Linguistics*.

Jia-Chen Gu, Chao-Hong Tan, Caiyuan Chu, Zhen-Hua Ling, Chongyang Tao, Quan Liu, and Cong Liu. 2023b. MADNet: Maximizing addressee deduction expectation for multi-party conversation generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7681–7692, Singapore. Association for Computational Linguistics.

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)*, pages 5086–5097, Dublin, Ireland. Association for Computational Linguistics.

Jia-Chen Gu, Chongyang Tao, Zhenhua Ling, Can Xu, Xiubo Geng, and Daxin Jiang. 2021. Mpc-bert: A pre-trained language model for multi-party conversation understanding. *ArXiv*, abs/2106.01541.

Wenpeng Hu, Zhangming Chan, Bing Liu, Dongyan Zhao, Jinwen Ma, and Rui Yan. 2019. Gsn: A graph-structured network for multi-party dialogues. In *International Joint Conference on Artificial Intelligence*.

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)*, pages 1909–1919, Hong Kong, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yiyang Li and Hai Zhao. 2023. EM pre-training for multi-party dialogue response generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 92–103, Toronto, Canada. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic. Association for Computational Linguistics.

Shikib Mehri, Evgeniia Razumovskaia, Tiancheng Zhao, and Maxine Eskenazi. 2019. Pretraining methods for dialog context representation learning. In *Proceedings of the 57th Annual Meeting of the Association for*

Computational Linguistics, pages 3836–3845, Florence, Italy. Association for Computational Linguistics.

Zhao Meng, Lili Mou, and Zhi Jin. 2017. Towards neural speaker modeling in multi-party conversation: The task, dataset, and models. In *International Conference on Language Resources and Evaluation*.

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*, pages 2133–2143, Austin, Texas. Association for Computational Linguistics.

Sunjeong Park and Youn-kyung Lim. 2020. Investigating user expectations on the roles of family-shared ai speakers. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA. Association for Computing Machinery.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Weizhou Shen, Junqing Chen, Xiaojun Quan, and Zhixiang Xie. 2020. Dialogxl: All-in-one xlnet for multi-party conversation emotion recognition. In *AAAI Conference on Artificial Intelligence*.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385, Minneapolis, Minnesota. Association for Computational Linguistics.

Weishi Wang, Steven C.H. Hoi, and Shafiq 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)*, pages 6581–6591, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Rui Zhang, Honglak Lee, Lazaros Polymenakos, and Dragomir 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 and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

A Data Statistics

Datasets	Train	Valid	Test
Ouchi and Tsuboi (2016)	461,120	28,570	32,668
Hu et al. (2019)	311,725	5,000	5,000

Table 5: Statistics of the two benchmarks evaluated in this paper.

B Case Study

Case 1

SPK1: e : could not open lock file FILEPATH - open (13 permission denied) e : unable to lock the administration directory (FILEPATH are you root ?
SPK2: you need to be using sudo
SPK3: ah , ok , so it 's not just me
SPK4: you can read all about FILEPATH issues on URL
SPK2: you can edit off the us . part of the url and that should fix you up

target-utterance: e : could not open lock file FILEPATH - open (13 permission denied) e : unable to lock the administration directory (FILEPATH are you root ?
answer speaker: SPK5
answer adressee: SPK1

Generated Responses:

Models	Response
Gold Label	well , you are n't root , are n't you ? EMOJI
Bart	are you using sudo or gksudo ?
HeterMPC	please don't flood, use url to paste, do nix, do not use enter as punctuation.
MADNet	please don't flood, use URL to paste, do nt use enter as punctuation.
SS-MPC	you need to be able to open it as root

Case 2

SPK1: many iconpacks may be for the older kde3
SPK2: i have run checkdisk from windows and i have many errors in partition
SPK3: sure , ai n't there a way to filter out 4.1 packs ?
SPK1: thats not a good sign .
SPK3: i 'm using kde to do that

target-utterance: sure , ai n't there a way to filter out 4.1 packs ?
answer speaker: SPK1
answer adressee: SPK3

Generated Responses:

Models	Response
Gold Label	i imagine it depends on FILEPATH you are looking
Bart	i 'm not sure , but i guess it 's filter by version
HeterMPC	i don't know, i'm not sure how to do it
MADNet	i don't know, i've never used kde.
SS-MPC	there is a way , but i do n't remember the kde3 way

Case 3

SPK1: shaft : not now , only with the ubuntu cd i can see the disk unpartitioned
SPK2: see when i stopped a linucx distro during install a while back everything refused to boot from cd until i sorted it out
SPK1: i gues is domething like that . how do i do that ?
SPK2: well its just a guess from previous experience ,
SPK2: id try installing ubuntu , with it using all your hdd , then try installing windows over it , thats what i had to do
SPK1: you mean , i will format the disk again with windows , errasing the previous ubuntu installing ?

target-utterance: you mean , i will format the disk again with windows , errasing the previous ubuntu installing ?
answer speaker: SPK2
answer adressee: SPK1

Generated Responses:

Models	Response
Gold Label	going off my experience with what seemed the same problem yes , unless you can ask someone else that may have a technical solution for you EMOJI
Bart	i will format the disk again with windows , and erase the previous ubuntu install
HeterMPC	i think you can do that with the ubuntu cd
MADNet	i dont know, i dont use windows, so i dont know
SS-MPC	you should be able to use the entire drive again , but i dont know if it will work

C Training and Inference Time Comparison

	Time (Sec)		Average Token
Model	**Training**	**Inference**	**Length**
HeterMPC	6,707	677.08	16.52
SS-MPC$_{bart\_base}$	2,486	148.53	22.63
SS-MPC$_{bart\_large}$	6,592	265.78	26.59

Table 6: Training and Inferencing time comparison on HeterMPC and SS-MPC

Table 6 compares the training and inference times of HeterMPC, a previous MPC response generation model, with our proposed SS-MPC. For training, the comparison reflects the per-epoch training time for each model using a batch size of 8, while inference time represents the total time required for the model to inference all the test data set. All experiments were conducted on a single RTX 3090 GPU.

As shown in the table 6, SS-MPC has the faster training and inference times compared to HeterMPC. This is mainly due to the graph-based architecture used in HeterMPC, which incurs additional computational overhead.

Moreover, the SOTA model, MADNet, is the advanced version of HeterMPC by modifying graph structure and incorporating an EM algorithm, which results in even longer processing times.

Given these observations, it is more appropriate to compare HeterMPC and MADNet with SS-MPC based on the BART-large initialized version, rather than the BART-base.

D Addressee Prediction in Real-World MPC scenario

Model	**Utt. Info.**	**Target Utt.**	**Adr.**
GIFT (Gu et al., 2023a)		-	90.18%
SS-MPC$_{Encoder}$		76.38%	89.92%
SS-MPC$_{real\text{-}world}$	-last utt.	62.90%	78.40%

Table 7: Performance of predicting target-utterance of addressee in other MPC model. Unlike other models, SS-MPC Encoder does not utilize final response.

Table 7 resents the addressee prediction performance of the SS-MPC encoder in the real-world MPC scenario described in Chapter 4.4. Although the accuracy drops slightly compared to the original SS-MPC, it still hovers around 80%, suggesting that SS-MPC can predict the appropriate addressee and generate responses accordingly in realistic settings.

	Qwen2.5-3B	**SS-MPC$_{\text{Qwen2.5-3B}}$**
Fine-tuned	O	O
Dialogue Structuralization	X	O
Input Example		
Context	SpeakerA: Utterance1	[IDX_1] [IDX_none] [SPK_1] [SPK_none] Utterance1
	SpeakerB: Utterance2	[IDX_2] [IDX_1] [SPK_2] [SPK_1] Utterance2
	SpeakerC: Utterance3	[IDX_3] [IDX_2] [SPK_3] [SPK_2] Utterance3
	SpeakerA: Utterance4	[IDX_4] [IDX_3] [SPK_2] [SPK_2] Utterance1
Response	SpeakerB: Response	[IDX_5] [IDX_4] [SPK_2] [SPK_1] Utterance1

Table 8: Comparison between Qwen2.5-3B and SS-MPC$_{\text{Qwen2.5-3B}}$

Models	**BLEU-1**	**METEOR**	**ROUGE-L**
Qwen2.5-3B	33.62	31.89	33.09
SS-MPC$_{Qwen2.5-3B}$	**34.62**	**34.20**	**33.33**

Table 9: Applied result on LLM. We use Qwen2.5-3B model for training and inference.

E Adaptation on Large Language Models

The concept of dialogue structuralization is also applicable to Large Language Model (LLM). Table 9 shows the application of dialogue structuralization on Qwen2.5-3B. In the table, Qwen2.5-3B refers to the Qwen2.5-3B model **with finetuning** on the MPC response generation task, and SS-MPC$_{\text{Qwen2.5-3B}}$ refers to the Qwen2.5-3B model **finetuned with Dialogue Structuralization** on the same task. The details with structuralized input used for each model is marked in the table 8.

The Table 9 shows the effects of dialogue structuralization in LLMs. The results demonstrate that sequence-structured input significantly impacts performance. Especially, Qwen achieved an improvement of nearly 1%p in BLEU-4 and 2.5%p in METEOR solely by utilizing sequence-structured input. This highlights the importance of incorporating conversational structure into the input representation.

F Human Evaluation Instruction

The judges were asked to score a total of 100 samples, assigning either 0 or 1 for each of the following criteria: Relevance, Fluency, and Informativeness. The evaluation criteria for each category are as follows:

Relevance: Does the response actually address the main point of the question? In other words, is the answer appropriate to the intent of the question?

Fluency: Regardless of the question, is the response expressed in a natural and fluent manner?

Informativeness: (Considered together with the question) Does the response provide information that would be helpful to the user?

G License

The data used in this paper can be found on https://github.com/ryanzhumich/sirnn (Ubuntu IRC (2016)) and https://github.com/morning-dews/GSN-Dialogues (Ubuntu IRC (2019)) We utilize parts of the code provided by HeterMPC[6], which is licensed under the Apache 2.0 License.

[6] https://github.com/lxchtan/HeterMPC