Pre-training Multi-party Dialogue Models with Latent Discourse Inference

Yiyang Li[1,2,*], **Xinting Huang**[3,†], **Wei Bi**[3] and **Hai Zhao**[1,2,†]
[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University
[3] NLP Center, Tencent AI-Lab
eric-lee@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn
{timxinhuang,victoriabi}@tencent.com

Abstract

Multi-party dialogues are more difficult for models to understand than one-to-one two-party dialogues, since they involve multiple interlocutors, resulting in interweaving reply-to relations and information flows. To step over these obstacles, an effective way is to pre-train a model that understands the discourse structure of multi-party dialogues, namely, to whom each utterance is replying. However, due to the lack of explicitly annotated discourse labels in multi-party dialogue corpora, previous works fail to scale up the pre-training process by putting aside the unlabeled multi-party conversational data for nothing. To fully utilize the unlabeled data, we propose to treat the discourse structures as latent variables, then jointly infer them and pre-train the discourse-aware model by unsupervised latent variable inference methods. Experiments on multiple downstream tasks show that our pre-trained model outperforms strong baselines by large margins and achieves state-of-the-art (SOTA) results, justifying the effectiveness of our method. The official implementation of this paper is available at https://github.com/EricLee8/MPD_EMVI.

1 Introduction

Dialogue system is an important area that has been studied for a long time in natural language processing field. Different from plain texts, dialogues are harder for models to understand since they are full of informal, colloquial expressions, and many ellipses (Yang and Choi, 2019; Reddy et al., 2019; Li et al., 2022). Among them, multi-party dialogues are even more complex since they involve multiple interlocutors, resulting in interweaving reply-to relations and information flows (Gu et al., 2021; Sun et al., 2021; Gu et al., 2022b). Specifically, in multi-party dialogues, the current utterance can be a reply to any preceding utterance in the dialogue history, forming complex discourse structures.

Intuitively, it is important for models to perceive the discourse structures, or in other words, to whom each utterance is replying, when comprehending multi-party dialogues. This intuition is in line with the process we humans participate in multi-party dialogues: we first read or listen to the dialogue history, knowing who speaks what to whom, then choose an utterance as the addressee, and finally utter a response. Literature has also justified that incorporating the discourse knowledge into models is beneficial for better understanding multi-party dialogues (Li et al., 2020; Jia et al., 2020; Li and Zhao, 2021; Ma et al., 2022). Unfortunately, the process of choosing addressees is a naturally unobservable action, resulting in a large amount of multi-party conversational data without addressee labels. In this work, we focus on leveraging the unlabeled data to pre-train a model for multi-party dialogue understanding.

To utilize the discourse structure, previous works seek help from human laborers to annotate the addressee labels on small datasets, where they either explicitly model the discourse structure using Graph Neural Networks or multi-task learning (Hu et al., 2019; Sun et al., 2021; Li et al., 2021; He et al., 2021; Gu et al., 2022a), or attempt to pre-train a model using objectives that are related to addressees by supervised learning (Gu et al., 2021). These works heavily rely on annotated addressee labels, which are rare in practice since the annotation process requires large amounts of human resources. As a result, they fail to be practical in real-world applications and are hard to scale up by utilizing more unlabeled multi-party conversational data.

To make full use of the unlabeled corpora, a natural idea is to treat the unobservable discourse structure (reply-to relations) as latent variables, then adopt latent variable models to jointly infer them and optimize the discourse-aware models. How-

[*] Work done while interning at Tencent AI Lab.
[†] Corresponding author. This paper was partially supported by Key Projects of National Natural Science Foundation of China (U1836222 and 61733011).

ever, it is not that simple when it comes to practice. For the Expectation-Maximization (EM) algorithm, the posterior distribution of the reply-to relations is intractable since it requires a square-level time complexity. If we turn to Variational Inference (VI) for help, the choice of the categorical prior distribution of the reply-to relations becomes troublesome: naive assumptions such as uniform distributions are too weak to make the training process converge.

To step over the above obstacles, we subtly combine the single-turn EM algorithm and multi-turn VI into a two-stage pre-training strategy. In the first stage, we adopt the EM algorithm to jointly model the context-response matching objective and single-turn addressee inference, which requires only a linear time complexity and can preliminarily guide the model to a relatively good converging point with utterance-level knowledge. In the second stage, we extend the latent variables from single-turn addressees to multi-turn reply-to relations and optimize the model via both the EM algorithm and VI framework, where the prior distribution of the reply-to relations is no longer troublesome since it can be derived exactly from the E-steps. This stage further enhances the model with discourse-level knowledge and guides it converge to a better point.

To sum up, the contributions of this work are:

- We successfully scale up the pre-training for multi-party dialogue understanding by leveraging the huge amounts of multi-party conversational corpora without addressee labels, while previous methods fail to work on these corpora.
- We subtly combine the single-turn EM algorithm and multi-turn VI framework in a two-stage pre-training process, which equips the model with knowledge of different granularities and makes it converge to an ideal point.
- The pre-trained model serves as a powerful encoder for multi-party dialogues and outperforms strong baselines by large margins, achieving SOTA results on multiple downstream tasks.

2 Related Works

2.1 Multi-party Dialogue Modeling

Several works have studied the modeling of multi-party dialogues before. Hu et al. (2019) propose to encode the reply-to relations with Graph Structural Networks (GSN). They utilize the addressee annotations and speaker information in the dataset to construct discourse and speaker graphs, then adopt a backward-forward strategy to pass messages between utterances. Sun et al. (2021); Gu et al. (2022a) further extend the modeling from homogeneous graphs to heterogeneous graphs by utilizing the Relational Graph Convolutional Networks to encode the heterogeneous information. However, their solutions all require annotated addressee labels in the multi-party dialogue dataset, which are rare and expensive to obtain in real-world applications. On the contrary, our work requires no addressee annotations, which saves human labors and can be scaled up using large unlabeled corpora.

Most related to our work, Li and Zhao (2023) attempts to improve the response generation model for multi-party dialogues by employing the EM algorithm to infer single-turn addressees. However, their approach encounters limitations when it comes to expanding the pre-training process due to the slow generative E-steps. Additionally, their work fails to fully exploit the discourse structure of the dialogue history, as they solely focus on the single-turn addressees. In contrast, our method not only scales up the pre-training by employing faster objectives, but also extends the latent variables from single-turn addressees to multi-turn reply-to relations to enhance the model with discourse-level knowledge, which is more important in comprehending multi-party conversations.

2.2 Dialogue Pre-training

To bridge the gap between pre-trained language models (PLMs) on plain texts and dialogue texts, many attempts have been made to pre-train a model for dialogues. Bao et al. (2020); Chen et al. (2022b) treat the dialogue intent as discrete or continuous latent variables to pre-train a model that solves the one-to-many problem in dialogue response generation task. Mehri et al. (2019); Xu and Zhao (2021); Zhang and Zhao (2021) design different self-supervised objectives for two-party dialogue context modeling. Different from their two-party setting, our work focuses on the multi-party scenario, where the addressee information should be concerned. Gu et al. (2021) also consider pre-training a model for multi-party dialogue understanding. They pre-train their model on a small dataset with annotated addressee labels by supervised addressee-related objectives. Since annotations are required, their pre-training strategy fails to scale up by using the unlabeled data. In contrast, our method is labor-free since the addressees are inferred by unsupervised latent-variable methods.

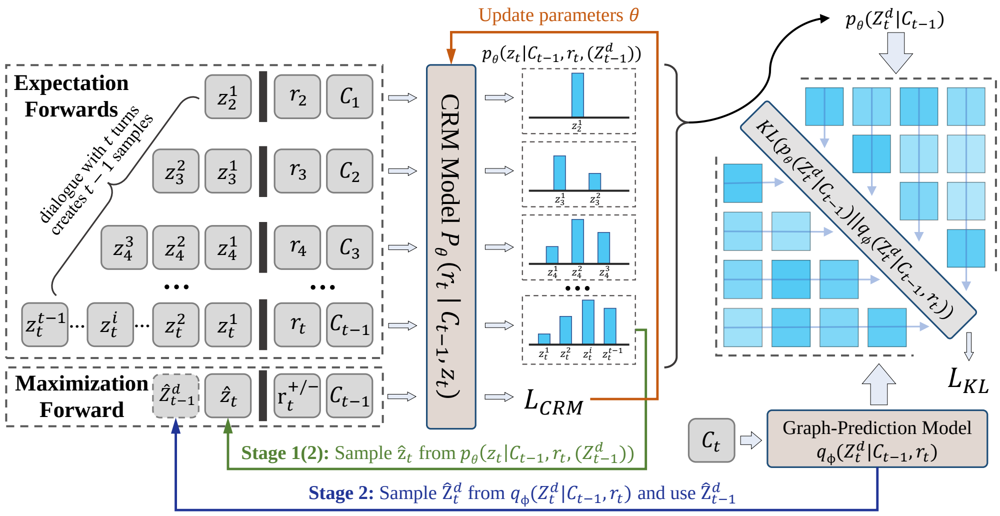

Figure 1: The overview of our pre-training process. The left part shows the turn-level Expectation-Maximization process while the right part illustrates the discourse-level Variational Inference enhancement.

3 Methodology

In general, Figure 1 illustrates the overview of the proposed two-stage pre-training strategy. The left part illustrates the single-turn Expectation-Maximization process, where we iteratively conduct E-steps to infer the latent addressee z_t (left-upper part and the green arrow), and M-steps to optimize the model via addressee-aware context-response matching (CRM) objective (left-lower part and the orange arrow). The right part illustrates the multi-turn Variational Inference process, which is incorporated into the EM framework in the second pre-training stage. We extend the latent variables from the single-turn addressees to multi-turn addressee-graphs, and jointly optimize the discourse-aware context-response matching model (the blue arrow) and the graph-prediction model q_ϕ by Variational Inference. In the next sections, we will introduce the two pre-training stages in detail.

3.1 Single-turn Addressee Inference

As mentioned in Section 1, simply applying the EM algorithm to infer all reply-to relations in the dialogue requires a square-level time complexity, which is intolerably time-consuming for the pre-training on large corpora. To solve this issue, we step back in the first pre-training stage to focus on the modeling and inference of single-turn addressees. For one thing, it requires only a linear time complexity for each training instance and hence can be optimized via the EM algorithm. For another, the addressee distributions output by the E-steps can derive the prior distribution of the reply-to relations, which can be utilized by the Variational Inference process in the second pre-training stage.

3.1.1 Preliminaries

Let's consider the process that humans participate in a multi-party dialogue in the t_{th} turn: we first read the dialogue history C_{t-1}, then choose an addressee utterance z_t that we want to reply, and finally utter a response sentence r_t. Formally, a multi-party dialogue corpus contains dialogues with format (C_{t-1}, z_t, r_t), where the annotations of z_t are lacking in most corpora. Here $C_{t-1} = \{\mathrm{S_1}\text{: } \mathrm{U_1}[\mathrm{SEP}]\mathrm{S_2}\text{: } \mathrm{U_2}[\mathrm{SEP}] \ldots \mathrm{S_{t\text{-}1}}\text{: } \mathrm{U_{t\text{-}1}}[\mathrm{SEP}]\mathrm{S_t}\}$, where $\mathrm{S_i}$ and $\mathrm{U_i}$ are the speaker and utterance of the i_{th} turn, respectively. Addressee $z_t \in [1, t-1]$ is a one-hot vector that indicates to whom we reply in the current turn t. In our settings, each utterance except the first one has exactly one addressee.

The conversation process can be formulated as $p_{\boldsymbol{\theta}}(r_t|z_t, C_{t-1})$, which models the probability of r_t being the correct response given C_{t-1} and z_t under trainable parameters $\boldsymbol{\theta}$. In large datasets without addressee labels z_t, we should infer the unobservable latent addressees. To this end, we adopt the EM algorithm to iteratively infer the addressees $p_{\boldsymbol{\theta}}(z_t|C_{t-1}, r_t)$ during the E-steps, and optimize the model $p_{\boldsymbol{\theta}}(r_t|z_t, C_{t-1})$ using the CRM objective during the M-steps.

3.1.2 Maximization Step

Suppose we have already obtained the inferred addressees from the E-step, two questions should be answered in the M-step: how to design the addressee-aware model architecture, and how to design the CRM task that enforces the model to leverage addressee information.

To answer the first question, our solution is straightforward but effective: similar to the speaker or turn embeddings in previous works (Gu et al., 2020; Zhang et al., 2021), we add an addressee embedding on top of the token and positional embeddings to indicate which utterance is the current addressee. Note that we have also tried other addressee modeling methods such as the prompt-based ones, yet they are not as effective as the addressee embeddings.

To answer the second question, we first follow the common practice to formulate the CRM task as a binary classification problem (Tao et al., 2021; Su et al., 2021), where the model should distinguish positive (correct) responses r_t^+ from the negative ones r_t^- in the current dialogue turn t. To make the CRM task more addressee-related, besides simple negatives that are randomly sampled from the whole training corpus, we also construct hard negatives that are sampled from the later ($> t$ turns) utterances in the same dialogue. Liu et al. (2019) point that simple negatives are easily distinguishable from positive ones by their topic differences. In other words, they can be predicted as negatives without the specified addressee information, which can not help the addressee inference process in the E-step. In contrast, the topic of each hard negative response is coherent with the current dialogue, making them hard to be classified with only the topic or sequential features. As a result, the model is forced to seek clues from the speaker and addressee information to distinguish those hard negatives, which greatly benefits the E-step.

With the model and training data at hand, we adopt binary cross-entropy loss as the objective function for the CRM task:

$$\begin{aligned}\mathcal{L}_{CRM} = -(y_t \times \log[\,p_{\boldsymbol{\theta}}(r_t|z_t, C_{t-1})\,] \\ + (1-y_t) \times \log[\,1 - p_{\boldsymbol{\theta}}(r_t|z_t, C_{t-1})\,])\end{aligned} \tag{1}$$

Here $y_t \in \{0, 1\}$ is the ground truth label that indicates whether r_t is a positive response. The left lower part and the orange arrow of Figure 1 illustrate the maximization step, where we ignore $\hat{Z}_{t-1}^d$ since it will be introduced in Section 3.2.

3.1.3 Expectation Step

The inference of latent addressees can be formulated as calculating $p_{\boldsymbol{\theta}}(z_t|C_{t-1}, r_t)$. In other words, given the dialogue history C_{t-1} and current response r_t, we should infer the posterior categorical distribution of the addressee $z_t \in [1, t-1]$. Consider the factorization of this posterior distribution:

$$\begin{aligned}p_{\boldsymbol{\theta}}(z_t|C_{t-1}, r_t) &= \frac{p_{\boldsymbol{\theta}}(C_{t-1}, z_t, r_t)}{p_{\boldsymbol{\theta}}(C_{t-1}, r_t)} \\ &= \frac{p_{\boldsymbol{\theta}}(C_{t-1}) \times p_{\boldsymbol{\theta}}(z_t|C_{t-1}) \times p_{\boldsymbol{\theta}}(r_t|z_t, C_{t-1})}{p_{\boldsymbol{\theta}}(C_{t-1}) \times p_{\boldsymbol{\theta}}(r_t|C_{t-1})} \\ &= \frac{p_{\boldsymbol{\theta}}(z_t|C_{t-1}) \times p_{\boldsymbol{\theta}}(r_t|z_t, C_{t-1})}{p_{\boldsymbol{\theta}}(r_t|C_{t-1})}\end{aligned} \tag{2}$$

where the factorization order of the numerator follows human habits when participating in a multi-party dialogue mentioned at the beginning of Section 3.1.1. In the denominator, $p_{\boldsymbol{\theta}}(r_t|C_{t-1})$ is irrelevant to z_t. In the numerator, we assume a uniform prior distribution $p_{\boldsymbol{\theta}}(z_t|C_{t-1})$, hence this term is also irrelevant to z_t. Hence, we can derive that:

$$p_{\boldsymbol{\theta}}(z_t|r_t, C_{t-1}) \propto p_{\boldsymbol{\theta}}(r_t|z_t, C_{t-1}) \tag{3}$$

Adopting this equation and the trained CRM model $p_{\boldsymbol{\theta}}(r_t|z_t, C_{t-1})$ from the M-step, we can now calculate the posterior distribution of z_t by traversing all possible addressees $\{z_t^i\}_{i=1}^{t-1}$:

$$p_{\boldsymbol{\theta}}(z_t^i|r_t, C_{t-1}) = \frac{p_{\boldsymbol{\theta}}(r_t|z_t^i, C_{t-1})}{\sum_{j=1}^{t-1} p_{\boldsymbol{\theta}}(r_t|z_t^j, C_{t-1})} \tag{4}$$

The left upper part and green arrow in Figure 1 shows the E-step, where we ignore Z_{t-1}^d since it will be introduced in Section 3.2.

3.2 Multi-turn Addressee-graph Inference

Once the EM iterations have reached a relatively good converging point, we dive into the second stage of training by additionally integrating the multi-turn Variational Inference task into the EM framework. This stage further enhances the model with discourse-level knowledge, making it possible to converge to a better point.

The discourse-level VI extends the latent variables from single-turn addressees z_t to multi-turn addressee-graphs $Z_t^d \in \mathcal{R}^{t \times t}$, which is an adjacent matrix indicating to which addressee each utterance is replying to. In other words, the model now should infer all the addressees of each utterance U_i in the dialogue context C_t. As mentioned in Section 3.1, adopting the EM algorithm to infer Z_t^d is intolerably time-consuming. To solve this issue, we borrow the idea of Variational Inference (Kingma and Welling, 2014) to adopt a graph-prediction model $q_{\boldsymbol{\phi}}(Z_t^d|C_{t-1}, r_t)$ with additional trainable parameters ϕ to predict the addressee-graphs. Formally, we maximize the log-likelihood of the observed data $\log p_{\theta}(r_t|C_{t-1})$ (conditioned on the dialogue history C_{t-1}) by improving its Evidence Lower Bound (ELBO):

$$\begin{aligned} &\mathrm{ELBO}(\theta, \phi; r_t, C_{t-1}) = \\ &\quad \mathbb{E}_{q_{\phi}(Z_t^d|r_t, C_{t-1})}[\log p_{\theta}(r_t|Z_t^d, C_{t-1})] \\ &\quad - D_{KL}(q_{\phi}(Z_t^d|r_t, C_{t-1}) \| p_{\theta}(Z_t^d|C_{t-1})) \end{aligned} \tag{5}$$

Three important distributions are presented in this equation. First, $p_{\theta}(r_t|Z_t^d, C_{t-1})$ is a new formulation of the CRM task, where single-turn addressees z_t now becomes multi-turn addressee-graphs Z_t^d. Second, $p_{\theta}(Z_t^d|C_{t-1})$ is the conditional prior distribution of latent variable Z_t^d under parameters θ. Finally, $q_{\phi}(Z_t^d|C_{t-1}, r_t)$ is the graph-prediction model, which predicts the edges from each response to its addressee by outputting the estimated posterior distribution of Z_t^d. Next, we introduce the modeling of these distributions in detail.

3.2.1 Discourse-aware CRM

Let's start with $p_{\theta}(r_t|Z_t^d, C_{t-1})$. Given the dialogue history C_{t-1} and the addressee-graph Z_t^d sampled from q_{ϕ}, we model the CRM task by imitating *careful* human readers: when we *seriously* reply to an utterance in a multi-party dialogue, instead of focusing solely on the current addressee utterance z_t itself, we tend to focus more on the utterances in the reply-chain of r_t, namely, the k-hop ancestors of r_t in the addressee-graph Z_t^d. Formally, we first extract the utterance representations of the k-hop ancestors of r_t to form a reply-chain information representation $H_t^k \in \mathcal{R}^{k\times d}$, then model $p_{\theta}(r_t|Z_t^d, C_{t-1})$ with an MLP.

To accelerate the computation of the k-hop ancestors, we construct a one-hot vector $a_t \in \mathcal{R}^{1\times t}$ to indicate the position of the current response r_t. Right-multiplying this vector by the addressee-graph matrix Z_t^d for i times yields the position vector of its i_{th} ancestor. $p_{\theta}(r_t|Z_t^d, C_{t-1})$ can now be formulated as follows:

$$\begin{aligned} &H_t^k = \mathrm{concat}[\{a_t(Z_t^d)^i\}_{i=0}^{k-1}] \cdot H_t^u \in \mathcal{R}^{k\times d} \\ &p_{\theta}(r_t|Z_t^d, C_{t-1}) = \sigma(\mathrm{MLP}_{\theta}(\mathrm{flatten}(H_t^k))) \end{aligned} \tag{6}$$

Here $\mathrm{concat}[\cdot]$ is concatenation, $\mathrm{flatten}$ means squeezing the matrix into a vector, $\mathrm{MLP}_{\theta} \in \mathcal{R}^{kd\times 1}$ is a linear projection and σ is the Sigmoid function. In this pre-training stage, $p_{\theta}(z_t|r_t, C_{t-1})$ and $p_{\theta}(r_t|z_t, C_{t-1})$ in the equations of Section 3.1 have now become $p_{\theta}(z_t|r_t, Z_{t-1}^d, C_{t-1})$ and $p_{\theta}(r_t|Z_t^d, C_{t-1})$, respectively. For more detailed proofs, please refer to Appendix A.

3.2.2 Conditional Prior Distribution

Then, we focus on the conditional prior distribution $p_{\theta}(Z_t^d|C_{t-1})$. The choice of the prior distribution is vital to the convergence of Variational Inference (Kingma and Welling, 2014; Chen et al., 2022a). Previous works either make strong assumptions over the prior distribution, like Uniform and Gaussian (Qian et al., 2022), or use additional annotation models to approximate the prior distribution (Chen et al., 2022a). However, as mentioned in Section 1, they fail to work in our scenario since naive assumptions are too weak to make the training process converge. Thanks to the EM training process, the prior distribution $p_{\theta}(Z_t^d|C_{t-1})$ can be derived exactly from the previous $t-1$ E-steps in this dialogue. Formally, it can be calculated as:

$$\begin{aligned} &E(i) = p_{\theta}(z_i|r_i, Z_{i-1}^d, C_{i-1}) \\ &p_{\theta}(Z_t^d|C_{t-1}) = \Pi_{i=1}^{t-1}[E(i)] \cdot U(|z_t|) \end{aligned} \tag{7}$$

Here $U(|z_t|)$ is a uniform distribution over the length of the candidates of z_t. Due to the page limit, we put the detailed derivations of this equation in Appendix B. This equation subtly combines the EM training framework and the VI process, which guides the model converge to a better point by incorporating accurate prior knowledge of the discourse-level addressee-graphs.

3.2.3 Graph-prediction Model

Finally, we end with the graph-prediction model $q_{\phi}(Z_t^d|C_{t-1}, r_t)$. To compute the edges between each utterance pair, we first apply mean pooling over the corresponding token representations of each utterance to get utterance-level representations $H_t^u \in \mathcal{R}^{t\times d}$. After that, we compute the score of each utterance pair being the response-addressee by an MLP with trainable parameters ϕ to get a scoring matrix $S^u \in \mathcal{R}^{t\times t}$. Finally, q_{ϕ} is calculated as follows:

$$q_{\phi} = \text{Gumbel-Softmax}(S^u + M^u) \tag{8}$$

Here $M^u \in \mathcal{R}^{t\times t}$ is a masking matrix with $-\infty$ values on its upper triangular part to mask invalid positions, since each utterance can only reply to its previous ones. We adopt Gumbel-Softmax relaxation to make the sampling of q_{ϕ} differentiable, following Jang et al. (2017); Maddison et al. (2017).

3.3 Pre-training Objectives

Besides utterance-level CRM and discourse-level graph prediction, we also design an addressee-aware masked language modeling (MLM) task to preserve the token-level knowledge, which is introduced in detail in Appendix C. To sum up, the overall training objective in the M-step is:

$$\mathcal{L} = \mathcal{L}_{CRM} + \alpha\mathcal{L}_{KL} + \beta\mathcal{L}_{MLM} \quad (9)$$

Here α and β are two hyper-parameters and are set to 0 at the first pre-training stage.

4 Experiments

In this section, we introduce the experimental settings and present the results on downstream tasks.

4.1 Pre-training Settings

For the pre-training data, we use the script of (Zhang et al., 2020) to download Reddit posts from 2005 to 2020 and extract multi-party conversations to create a pre-training corpus of 17,154,613 dialogues. Since the pre-training corpus is huge, we split it into trunks of data and perform EM iterations on each of them. For backbone models, we choose $\text{BERT}_{\text{base}}$ (Devlin et al., 2019) and $\text{ELECTRA}_{\text{large}}$ (Clark et al., 2020). The former takes 4 days to converge in 8 NVIDIA A100 GPUs and the latter takes 12 days. For more details about the pre-training, please see Appendix D.

4.2 Downstream Settings

To test the capability of our pre-trained model, we conduct experiments on four downstream tasks based on multi-party dialogues.

Discourse Parsing requires the model to parse the reply-to links (addressee-graphs) in a multi-party dialogue and classify their relation types at the same time. For this task, we adopt Molweni (Li et al., 2020) as the benchmark dataset and use the F1 score of graph-prediction (F1_{G}) and relation classification (F1_{RL}) as the evaluation metrics.

Successful New Entry Prediction is to predict whether a newcomer's message will be responded to by other participants in a multi-party dialogue, which is formulated as a binary classification task. For this task, we adopt SNEP (Wang et al., 2022) as the benchmark dataset and use Area Under Curve (AUC) and F1 score as the evaluation metrics.

Extractive Question Answering requires the model to extract an answer span from the dialogue context given a question. For this task, we also adopt Molweni as the benchmark and use Exact-Match (EM) and F1 score as the evaluation metrics.

Response Generation aims at generating an appropriate response given the speaker and a specified addressee in a multi-party dialogue. For this task, we adopt Ubuntu IRC dataset (Hu et al., 2019) as the benchmark dataset and use BLEU, METEOR, and ROUGE-L as the evaluation metrics.

For more details about the datasets (statistics, data sources, etc.), please refer to Appendix E.

During the fine-tuning process, we discard the graph-prediction model q_ϕ since our model no longer requires explicit discourse modeling thanks to the implicit discourse knowledge learn from the pre-training. In our experiments, we make task-specific designs for each downstream task to fully utilize the addressee embedding to lay emphasis on important utterances that are not necessarily addressees, hence we call it *Adaptation Model*. For more details about the task-specific designs, please refer to Appendix F. To test the universality and simplify the usage of our pre-trained model, experiments are also conducted where we discard the addressee embedding and use only the parameters that are exactly the same as BERT, hence we call it *Vanilla Model*. Following previous works (Li et al., 2020; Gu et al., 2021; Wang et al., 2022), we mainly conduct our experiments based on $\text{BERT}_{\text{base}}$.

In Table 1, MPC-BERT (Gu et al., 2021) is introduced in Section 2.2, which is pre-trained on a small dataset with annotated addressee labels using supervised learning. BERT+CRM is an ablation model that is pre-trained using only the first stage (but with full data), which means only the CRM loss and EM training are adopted. +MLM means addressee-aware MLM objective is further added in the pre-training process and +VI represents our full model with two-stage pre-training. To study whether two-party dialogue models can still work in the multi-party scenario, we also conduct experiments on SPIDER-BERT (Zhang and Zhao, 2021), which is a model pre-trained on two-party dialogues using self-supervised objectives.

4.3 Experimental Results

We can see from Table 1 that our full model (+VI) significantly outperforms $\text{BERT}_{\text{base}}$ and MPC-BERT on all tasks, justifying the effectiveness of discourse knowledge modeling by incorporating VI into the EM training framework with two-stage pre-training. Besides, BERT+CRM is already strong

Model	Discourse Parsing		SNEP-Reddit		SNEP-Twitter		Extractive Q.A.	
	$F1_{RL}$	$F1_G$	AUC	F1	AUC	F1	EM	F1
Adaptation Model								
BERT-base	61.06	87.33	63.89	33.73	81.50	88.25	47.78	61.77
SPIDER-BERT	62.79	87.92	64.88	34.02	81.98	88.87	48.69	62.79
MPC-BERT	63.91	89.12	65.08	34.12	82.56	89.05	47.29	61.72
BERT+CRM	63.08	88.40	67.06	36.77	83.61	89.22	49.66	63.31
+MLM	63.79	88.42	67.32	36.58	83.72	89.33	50.03	63.54
+VI	**64.97**	**90.31**	**68.16**	**36.97**	84.06	**89.62**	51.17	**64.89**
Vanilla Model								
BERT-base	60.71	87.45	63.44	32.57	81.33	87.85	46.81	60.20
SPIDER-BERT	62.32	87.68	64.72	33.32	81.78	88.75	47.68	61.16
MPC-BERT	63.19	88.75	65.26	34.63	81.82	88.83	46.84	60.11
BERT+CRM	62.95	88.17	67.15	35.88	82.91	89.11	47.58	61.74
+MLM	63.19	88.91	67.16	36.36	83.48	88.92	47.51	62.43
+VI	64.22	89.59	68.09	36.96	**84.78**	89.61	**51.31**	64.52
ELECTRA-large	63.35	90.21	66.59	35.97	83.16	88.78	57.41	70.97
ELECTRA-our	**66.59**	**91.78**	**70.12**	**39.38**	**84.95**	**89.83**	**58.13**	**72.54**

Table 1: Results on classification-style downstream tasks.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
BERT	10.90	3.85	1.69	0.89	4.18	9.80
GSN	10.23	3.57	1.70	0.97	4.10	9.91
$\text{HeterMPC}_{\text{BERT}}$	**12.61**	4.55	2.25	1.41	4.79	11.20
BERT-our	11.78	**4.74**	**2.71**	**1.96**	**5.09**	**11.21**

Table 2: Results on the Ubuntu IRC benchmark.

enough to outperform MPC-BERT or to achieve comparable results, demonstrating the importance of scaling up the pre-training by EM algorithm and incorporating turn-level addressee knowledge. Also, adding addressee-aware MLM adds to performance gains, yet relatively slight. Finally, SPIDER-BERT performs relatively worse than multi-party models, which indicates the significance of designing models and objectives that are specific for multi-party dialogues. For more analyses about why the two-party objectives fail to work on the multi-party scenario, please refer to Appendix G.

Another observation is that the performance drops of the *Vanilla Model* compared with *Adaptation Model* is relatively minor on all dataset, which means it remains powerful even without the task-specific designs. This observation demonstrates that the discourse knowledge is indeed learned and stored in our pre-trained model.

Besides $\text{BERT}_{\text{base}}$, we also experiment with $\text{ELECTRA}_{\text{large}}$ to investigate whether our method can still enhance stronger PLMs. In this experiment, we compare the original $\text{ELECTRA}_{\text{large}}$ and our full model under the setting of *Adaptation Model*. As shown in the lower part of Table 1, our model outperforms $\text{ELECTRA}_{\text{large}}$ by large margins. This observation reveals that even strong PLMs, such as $\text{ELECTRA}_{\text{large}}$, still lack the knowledge to well understand multi-party dialogues, while our method can effectively enhance them by leveraging the discourse information inferred from the unlabeled datasets.

Our model can also improve the performance of response generation by enhancing the encoder side. Table 2 presents the results on the Ubuntu IRC dataset, where GSN (Hu et al., 2019) and HeterMPC (Gu et al., 2022a) utilize the discourse annotations in this dataset to explicitly model the reply-to relations by constructing homogeneous or heterogeneous graph neural networks. In contrast, the annotations are not used by our model since it is able to implicitly capture the reply-to information by the discourse knowledge learned during pre-training. As shown in Table 2, our model outperforms previous models even under the condition that we do not use additional annotations, demonstrating the strong capability of our model to understand the discourse structures.

5 Analyses

In this section, we make in-depth analyses to investigate more insights from our method.

5.1 Ablation Study

Since our model is trained on massive amounts of data, a natural question is whether the performance gains are from just seeing more conversations. To investigate this, we conduct experiments by remov-

Model	Molweni		SNEP-Twitter	
	$\mathrm{F1_{RL}}$	$\mathrm{F1_G}$	AUC	F1
Adaptation				
$\mathrm{BERT_{+CRM}}$	63.08	88.40	83.61	89.22
w/o EM	61.35	87.69	81.59	88.19
$\mathrm{BERT_{+CRM+MLM}}$	63.79	88.42	83.72	89.33
w/o EM	61.79	88.04	82.02	88.23
Vanilla				
$\mathrm{BERT_{+CRM}}$	62.95	88.17	82.91	89.11
w/o EM	61.42	88.04	81.45	88.57
$\mathrm{BERT_{+CRM+MLM}}$	63.19	88.91	83.48	88.92
w/o EM	61.73	88.34	82.12	88.25

Table 3: Ablation results on the Discourse Parsing (Molweni) and SNEP-Twitter task.

Model	Reddit	Molweni
$\mathrm{BERT_{base}}$	74.62	71.94
$\mathrm{ELECTRA_{large}}$	78.71	74.78

Table 4: F1 scores of zero-shot link prediction task.

ing the addressee-aware EM training process and only performing normal CRM and MLM on the full data. Also to test the out-of-domain generalization ability of our model, for this ablation experiment, we choose SNEP-Twitter and Discourse Parsing tasks since their data sources (Twitter and Ubuntu) are different from our pre-training source (Reddit).

Table 3 shows the ablation results, where we observe sharp performance drops when removing the EM training. This observation demonstrates the strong robustness and transferability of our model in out-of-domain data, thanks to the addressee knowledge learned from the EM process.

5.2 Zero-shot Graph-Prediction

To investigate to what extent the discourse knowledge is learned by our model, we test the zero-shot graph-prediction task on both Reddit and Molweni datasets. Note that during the pre-training stage, our model is trained on the pseudo-addressee-graphs that are inferred from the unlabeled dataset, hence we call this experiment zero-shot. Table 4 shows the $\mathrm{F1_G}$ scores of both datasets, where we observe good in-domain performance in Reddit and out-of-domain generalizability in Ubuntu (the Molweni dataset).

5.3 Addressee Distribution Shifts

At the beginning of our pre-training process, there are no annotated addressee labels in the training corpus, and the initial model is too weak to infer reasonable addressees using Eq. (4). To cold-start the EM bootstrapping process, we simply set the addressee of every response to be the last utterance

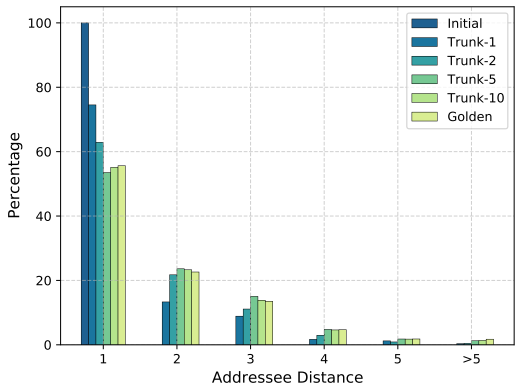

Figure 2: Distribution shift of addressee prediction.

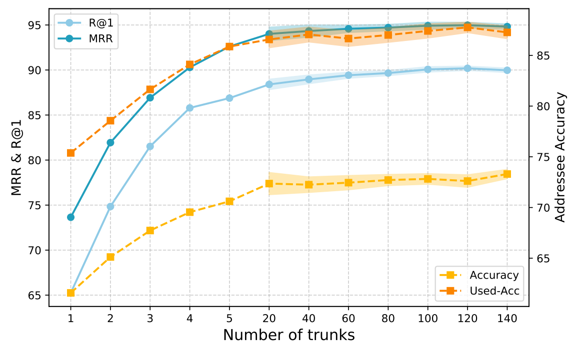

Figure 3: CRM scores vs. Addressee prediction accuracy during the pre-training process, where Used-Acc is the accuracy of top 50% confident samples that are used in the next M-step.

in the dialogue history (i.e., U_{t-1}), then perform the first round of M-step. This cold-start approach is different from, and much simpler than Li and Zhao (2023), where they utilize a trained discourse parser to label the addressees for the first M-step.

This strategy is simple but exhibits surprisingly good convergence: the distribution of the inferred addressees shifts from one-hot (the initial distribution) to a distribution that is close to the real addressee distribution in an annotated validation set, just after a few trunks. Figure 2 illustrates the distribution shift, where we draw the validation addressee distance distribution of the last E-step on each trunk. At the initial point, the addressees are all set to the last utterance, hence the percentage of addressees with distance 1 is 100%. With the increase of truck numbers, the addressee distance distribution gradually shifts and becomes closer and closer to the real distribution.

5.4 Pre-training Trending

Figure 3 illustrates the trending of both CRM scores (MRR and Recall@1) and addressee prediction accuracy of ELECTRA$_{\text{large}}$ during the pre-training process. After the 10_{th} trunk (the second pre-training stage), we compute the average and standard deviation over the ± 10 trunks of the index and show them in the figure as lines and shades.

First, we can see that both metrics grow together and mutually, which indicates with a stronger CRM model comes better addressee prediction accuracy, demonstrating the correctness of Eq. (3). Besides, the first stage of training reaches its convergence at around the 10_{th} trunk, by further incorporating VI at this point, both metrics keep growing and reach their top at around the 120_{th} trunk. Finally, the standard deviation is large at the beginning of the second stage of pre-training but gradually decreases with the convergence of the model.

6 Conclusion

In this paper, we point out that the lack of annotated addressee labels hinders the scaling-up of multi-party dialogue pre-training. To overcome this obstacle, we propose to utilize the unlabeled datasets by combining the EM algorithm and Variational Inference to jointly infer the discourse labels and pre-train the model with discourse-aware objectives on different granularities. Experimental results and extensive analyses have justified the effectiveness and transferability of our model on multiple downstream tasks.

Limitations

Despite the contributions of our work, there are also unavoidable limitations of it.

First, our method is based on the setting that each utterance in the dialogue except the first one has exactly one addressee. This setting holds tightly in online forums such as Twitter or Reddit, yet has its limit in group chats or meetings, where an utterance can reply to multiple or no addressees. However, this scenario is relatively rare in multi-party conversations. Considering this scenario is challenging and complicated since the one-to-many reply-to relations can cause the single-turn EM algorithm intractable. For this part, we leave it to future works.

Second, the Ubuntu IRC benchmark of response generation task is extracted from the Ubuntu Chat Corpus (Lowe et al., 2015), where people discuss the technical issues on the Ubuntu operating system. Due to the lack of human annotators with knowledge of Linux and Ubuntu, we do not conduct human evaluations on this dataset. However, we do provide the generated responses in our supplementary materials for those who are interested in the human evaluations.

References

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: Pre-trained dialogue generation model with discrete latent variable. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 85–96, Online. Association for Computational Linguistics.

Jiangjie Chen, Qiaoben Bao, Changzhi Sun, Xinbo Zhang, Jiaze Chen, Hao Zhou, Yanghua Xiao, and Lei Li. 2022a. LOREN: logic-regularized reasoning for interpretable fact verification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10482–10491. AAAI Press.

Wei Chen, Yeyun Gong, Song Wang, Bolun Yao, Weizhen Qi, Zhongyu Wei, Xiaowu Hu, Bartuer Zhou, Yi Mao, Weizhu Chen, Biao Cheng, and Nan Duan. 2022b. DialogVED: A pre-trained latent variable encoder-decoder model for dialog response generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4852–4864, Dublin, Ireland. Association for Computational Linguistics.

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

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.

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. ACM.

Jia-Chen Gu, Chao-Hong Tan, Chongyang Tao, Zhen-Hua Ling, Huang Hu, Xiubo Geng, and Daxin Jiang.

2022a. 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, and Zhen-Hua Ling. 2022b. Who says what to whom: A survey of multi-party conversations. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5486–5493. ijcai.org.

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. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3682–3692, Online. Association for Computational Linguistics.

Yuchen He, Zhuosheng Zhang, and Hai Zhao. 2021. Multi-tasking dialogue comprehension with discourse parsing. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 551–561, Shanghai, China. Association for Computational Linguistics.

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. ijcai.org.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Qi Jia, Yizhu Liu, Siyu Ren, Kenny Zhu, and Haifeng Tang. 2020. Multi-turn response selection using dialogue dependency relations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1911–1920, Online. Association for Computational Linguistics.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

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

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

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

Yiyang Li and Hai Zhao. 2023. Em pre-training for multi-party dialogue response generation.

Yiyang Li, Hai Zhao, and Zhuosheng Zhang. 2022. Back to the future: Bidirectional information decoupling network for multi-turn dialogue modeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2761–2774, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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.

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2022. Structural characterization for dialogue disentanglement. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 285–297, Dublin, Ireland. Association for Computational Linguistics.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2912–2924, Dublin, Ireland. Association for Computational Linguistics.

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

Yixuan Su, Deng Cai, Qingyu Zhou, Zibo Lin, Simon Baker, Yunbo Cao, Shuming Shi, Nigel Collier, and Yan Wang. 2021. Dialogue response selection with hierarchical curriculum learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1740–1751, Online. Association for Computational Linguistics.

Yang Sun, Nan Yu, and Guohong Fu. 2021. A discourse-aware graph neural network for emotion recognition in multi-party conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2949–2958, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chongyang Tao, Jiazhan Feng, Rui Yan, Wei Wu, and Daxin Jiang. 2021. A survey on response selection for retrieval-based dialogues. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4619–4626. ijcai.org.

Lingzhi Wang, Jing Li, Xingshan Zeng, and Kam-Fai Wong. 2022. Successful new-entry prediction for multi-party online conversations via latent topics and discourse modeling. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1663–1672. ACM.

Yi Xu and Hai Zhao. 2021. Dialogue-oriented pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2663–2673, Online. Association for Computational Linguistics.

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

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*, pages 270–278, Online. Association for Computational Linguistics.

Zhenyu Zhang, Tao Guo, and Meng Chen. 2021. Dialoguebert: A self-supervised learning based dialogue pre-training encoder. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 3647–3651. ACM.

Zhuosheng Zhang and Hai Zhao. 2021. Structural pre-training for dialogue comprehension. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5134–5145, Online. Association for Computational Linguistics.

A Derivation of E-step in Stage-2

In the second stage, the maximization step becomes the modeling of $p_{\boldsymbol{\theta}}(r_t|Z_t^d, C_{t-1})$, and the expectation step becomes computing the posterior distribution of $p_{\boldsymbol{\theta}}(z_t|r_t, Z_{t-1}^d, C_{t-1})$, accordingly. We also factorize this posterior distribution and omit $\boldsymbol{\theta}$ for simplicity:

$$
\begin{aligned}
p(z_t|r_t, Z_{t-1}^d, C_{t-1}) &= \frac{p(z_t, C_{t-1}, Z_{t-1}^d, r_t)}{p(C_{t-1}, r_t, Z_{t-1}^d)} \\
&= \frac{p(C_{t-1})\, p(Z_{t-1}^d|C_{t-1})\, p(r_t, z_t|C_{t-1}, Z_{t-1}^d)}{p(C_{t-1}) p(Z_{t-1}^d|C_{t-1})\, p(r_t|C_{t-1}, Z_{t-1}^d)} \\
&= \frac{p(z_t|C_{t-1}, Z_{t-1}^d)\, p(r_t|C_{t-1}, Z_{t-1}^d, z_t)}{p(r_t|C_{t-1}, Z_{t-1}^d)}
\end{aligned} \tag{10}
$$

In this equation, the factorization also follows human habit when we *seriously* participate in a multi-party dialogue: we first read the dialogue history (C_{t-1}), then analyze the discourse structure (reply-chains) of it ($Z_{t-1}^d|C_{t-1}$), then choose an addressee utterance we want to reply ($z_t|Z_{t-1}^d, C_{t-1}$), and finally utter a response to it ($r_t|z_t, Z_{t-1}^d, C_{t-1}$). In the last row of this equation, the denominator is irrelevant to z_t, and we also assume uniform distribution of $p(z_t|C_{t-1}, Z_{t-1}^d)$ in the numerator, which is also irrelevant to z_t. At this point, we can derive that:

$$
p(z_t|r_t, Z_{t-1}^d, C_{t-1}) \propto p(r_t|z_t, Z_{t-1}^d, C_{t-1}) \tag{11}
$$

and calculate the posterior distribution of z_t by traversing all possible addressees $\{z_t^i\}_{i=1}^{t-1}$:

$$
p(z_t^i|r_t, Z_{t-1}^d, C_{t-1}) = \frac{p(r_t|z_t^i, Z_{t-1}^d, C_{t-1})}{\sum\limits_{j=1}^{t-1} p(r_t|z_t^j, Z_{t-1}^d, C_{t-1})} \tag{12}
$$

B Derivation of Prior Distribution

We now derive how to compute the conditional prior distribution $p_{\boldsymbol{\theta}}(Z_t^d|C_{t-1})$, where we also omit $\boldsymbol{\theta}$ for simplicity. Firstly, we have

$$
\begin{aligned}
p(Z_t^d|C_{t-1}) &= p(z_t, Z_{t-1}^d|C_{t-1}) \\
&= p(Z_{t-1}^d|C_{t-1})\, p(z_t|C_{t-1}, Z_{t-1}^d)
\end{aligned} \tag{13}
$$

Here $p(z_t|C_{t-1}, Z_{t-1}^d)$ is assumed to be a uniform distribution in Appendix A, so we have:

$$
p(z_t|C_{t-1}, Z_{t-1}^d) \sim U(|z_t|) \tag{14}
$$

where $|z_t|$ is the length of the candidates of z_t. We now focus only on $p(Z_{t-1}^d|C_{t-1})$. Let's note $E(t) = p(z_t|r_t, Z_{t-1}^d, C_{t-1})$, we have:

$$
\begin{aligned}
&p(Z_{t-1}^d|C_{t-1}) \\
&= p(z_1, z_2, \ldots, z_{t-1}|C_{t-1}) \\
&= p(z_1|C_{t-1}) \ldots p(z_{t-1}|z_1, \ldots z_{t-2}, C_{t-1}) \\
&= \Pi_{i=1}^{t-1}\, p(z_i|Z_{i-1}^d, C_{t-1}) \\
&= \Pi_{i=1}^{t-1}\, p(z_i|Z_{i-1}^d, C_i) \\
&= \Pi_{i=1}^{t-1}\, p(z_i|r_i, Z_{i-1}^d, C_{i-1}) \\
&= \Pi_{i=1}^{t-1}\, [E(i)]
\end{aligned} \tag{15}
$$

In this equation, we use an intuitive constrain that $p(z_i|Z_{i-1}^d, C_{\geq i}) = p(z_i|Z_{i-1}^d, C_i)$ and $t-1 \geq i$, since in real-world scenario, we can not see the future dialogue contexts. Combining Eq. (14) and (15), we get:

$$
p_{\boldsymbol{\theta}}(Z_t^d|C_{t-1}) = \Pi_{i=1}^{t-1}[E(i)] \cdot U(|z_t|) \tag{16}
$$

which is exactly the same as Eq. (7).

C Masked Language Modeling Details

For addressee-aware masked language modeling (MLM) object described in Section 3.3, the three kinds of special words are masked with a higher probability. Specifically, for normal words, we mask them with a probability of 15%, for special words, the probability is 60%. The special words are randomly masked first. If the total masking ratio is over 30%, we randomly cancel some masks to reduce it below 30%. If the total masking ratio is below 15%, we repeat the masking process on those normal words to make the final masking ratio from 15% to 30%.

D Pre-training Details

As mentioned in Section 4.1, we split the pre-training data into several trunks and perform EM iterations on each of them. In our experiment, each trunk contains 600,000 (C_{t-1}, $r_t^{+/-}$) pairs and the total number of trunks is 158.

We perform 3 EM iterations for each trunk. At the end of each trunk, we will load data from the next trunk and perform E-step to infer the initial addressees for the first M-step of the next trunk. Note that the addressee initialization of the first trunk is a heuristic that sets the addressees of all response to the last utterance in the dialogue history, which is mentioned in Section 5.3.

After each E-step, we do not use all the training samples for the next M-step. Instead, we pick the samples with top 50% addressee prediction confidence scores for the next round of M-step. The confidence score is hard to design since simply adopting the highest probability calculated by Eq. (4) will cause length bias: dialogues with shorter context length will have larger highest probability. To solve this issue, we adopt two normalizing methods to normalize the logits output by the model to the same scale, and use the difference between the largest logits and the second largest logits $\max - \text{second\_max}$ to indicate the confidence level. Specifically, the two normalizing methods are min-max normalizing and average normalizing, respectively:

$$\begin{aligned} s_i^{\text{min-max}} &= \frac{s_i - min(S)}{max(S) - min(S)} \\ s_i^{\text{average}} &= \frac{s_i - min(S)}{avg(S)} \end{aligned} \tag{17}$$

Here $S = \{s_i\}_{i=1}^{t-1}$ is the logits scores output by the model. For each E-step, we compare the addressee prediction accuracy of the top 50% samples of both normalizing methods in the validation set, then choose the higher one as the normalizing method to select samples for the next round of M-step in the training set.

To preserve the knowledge learned from the previous trunks and meanwhile fully utilize the newly inferred addressees in each E-step, we remain the parameters of the PLM unchanged and re-initialize the parameters of the addressee embeddings and CRM classifier after each E-step. For the second pre-training stage, we also keep the parameters of the graph-prediction model unchanged.

We start the second stage of pre-training when the vanilla EM algorithm comes to its convergence. Specifically, when the addressee prediction accuracy stops to increase for continuous three trunks, we consider the EM iterations have converged and start the second stage of training by enabling the KL loss and switch the CRM model to the discourse-aware version. In our experiment, the EM algorithm converges at around the 10_{th} trunk. In the second stage of pre-training, the hyper-parameters in Eq. (9) are set to $\alpha = 1.0$ and $\beta = 0.5$, respectively.

We adopt Simulated Annealing during the Variation Inference to make the pre-training process stable and converge better. Specifically, the temperature coefficient τ of Eq. (8) is set to a high value (10.0) at the beginning of the second pre-training stage, then gradually decreases 0.1 with the graph-prediction model getting stronger and stronger. Formally, in the i_{th} trunk of the second pre-training stage, τ is calculated as $\tau = \max(0.1, \frac{1}{n-0.9})$.

	Train	Dev	Test	Total
# of Dialogues	8,771	883	100	9,754
# of Utterances	77,374	7,823	845	86,042
# of Questions	24,682	2,513	2,871	30,066

Table 5: Statistic of Molweni dataset.

	Twitter	Reddit
# of Dialogues	37,339	69,428
# of Utterances	179,265	236,764
# of Questions	29,340	12,199
# of Successful Entries	24,682	2,513
# of Failed Entries	7,999	57,229

Table 6: Statistic of SNEP dataset.

E Dataset Details

Molweni is a multi-party dataset for both discourse parsing and question answering tasks. It is sampled from the Ubuntu Chat Corpus (Lowe et al., 2015) and is annotated with question-answer pairs and discourse relations (reply-to links and edge types). This dataset contains multi-party dialogues discussing technical issues on the Ubuntu System, hence its topic and domain are very different from our pre-training corpus Reddit. Despite this, our model still generalizes well on this dataset by outperforming the baseline models by large margins. Table 5 shows the statistics of the Molweni dataset, where each utterance is annotated with its addressee and the relation type, each dialogue is annotated with several questions.

Successful New Entry Prediction (SNEP) is a multi-party dialogue dataset taken from Reddit and Twitter posts. This task is to predict whether a newcomer's message will be replied to by other users in a multi-party dialogue. This task would be an important part of the research in online assistants and social media. Table 6 shows the statistics of the SNEP dataset, where Reddit and Titter are two subsets.

Ubuntu IRC Benchmark is a dataset for multi-party dialogue response generation task. This dataset is also from the Ubuntu Chat Corpus (Lowe et al., 2015) and contains annotated addressee labels for each utterance. The generation task is formulated as follows: given the dialogue history and a specified addressee, the model should generate an appropriate response that is well related to the addressee. This dataset contains around 380,000 dialogues in total. For developing and testing set, there are 5,000 dialogues, respectively. For the evaluation scripts to compute ROUGE, METEOR, and BLEU, we use the same script as (Gu et al., 2022a).

F Adaptation Model Details

To make full use of the pre-trained addressee embedding, we design task-specific adaptation method for each downstream task.

For discourse parsing, the use of addressee embedding happens after the reply-to links are predicted. For each reply-to link, we model the addressee (the utterance that is pointed by another) with the addressee embedding and perform the relation classification.

For successful new entry prediction, we infer the addressee of the response to be studied (to predict whether it is a successful new entry) and adopt the addressee embedding to encode the dialogue. We perform mean pooling over the tokens of the response to get a vector, then adopt a binary classifier to make the final prediction.

For extractive question answering, we treat the question ans "response" and the utterance that contains the final answer (key-utterance) span as "addressee". Specifically, during training, we construct key-utterance labels with the annotated answer span and add an auxiliary key-utterance prediction module to predict the key-utterances. We adopt teacher forcing to model the answer span prediction task with the guidance of ground-truth key-utterance information by indicating the key-utterance with the addressee embedding. During inference, we first infer the key-utterance by the key-utterance prediction module, then use the predicted ones to model the answer span prediction task.

G Failure of Two-party Objectives

Let's take some common objectives of two-party dialogue pre-training for example.

First, consider the Utterance Order Restoration (UOS) objective that aims to restore the order of permutated utterances in two-party dialogues, or similarly the Utterance Swap Detection (USD) objective that determines whether there exists swapped utterances in the context. In multi-party dialogues, the order of two utterances that reply to the same root-utterance can be swapped, making these two objective inapplicable.

Second, consider the Utterance Restoration and Response Generation/Selection objectives, where the former restores masked utterance tokens using MLM and the latter generates or selects the ground truth response. These objectives can be too difficult for the model to learn without addressee information, due to the one-to-many problem of response-to-context when given different addressees.

The key motivation of this paper and the most difficult part of adopting self-supervised learning on multi-party dialogue is the lack of addressee information, which is subtly addressed by our EM+VI pre-training approach.

ACL 2023 Responsible NLP Checklist

A For every submission:

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

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

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

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

B ☑ Did you use or create scientific artifacts?

Section 3 and 4.

☑ B1. Did you cite the creators of artifacts you used?
Section 4.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
They are all publicly available and the license are available on GitHub.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 3 and 4.

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

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

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

C ☑ Did you run computational experiments?

Section 4 and 5.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4.

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

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

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

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

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

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.