Generic Dependency Modeling in Multi-Party Conversation

Anonymous ACL submission

Abstract

Modeling the dependency between utterances in a multi-party conversation facilitates the understanding of conversation more precisely and holistically. In this paper, we propose a simple and generic framework for this purpose, in which the dependency is built on discourse parsing of utterances. Particularly, we present two approaches to encoding the dependency, namely absolute dependency encoding and relative dependency encoding, and combine them in Transformers by modifying the computation of self-attention. To enhance the understanding of utterance dependency, we further introduce a span distance prediction pre-training task for the proposed model. Experimental results on four multi-party conversation benchmarks for different tasks show that this model successfully boosts the generic performance of Transformer-based language models. Systematic studies are conducted to investigate why utterance dependencies are essential for multi-party conversation tasks and how they are learned in a simple and effective framework.

1 Introduction

Most current research on dialog systems focuses on interactions between two interlocutors such as a human user and a conversational agent. There has been a strong need for extending it to multi-party conversations since numerous scenarios such as group meetings (Carletta et al., 2005) and online chat room (Uthus and Aha, 2013; Lowe et al., 2015) naturally involve more than two interlocutors (Traum, 2003). However, it tends to be more challenging to model multi-party conversations as there could be multiple threads (topics) ongoing and complicated dependency existing between utterances. As illustrated in Figure 1, multi-party conversations often face the challenge of referential ambiguity and lack coherence between consecutive utterances (Li et al., 2020; Jia et al., 2020).

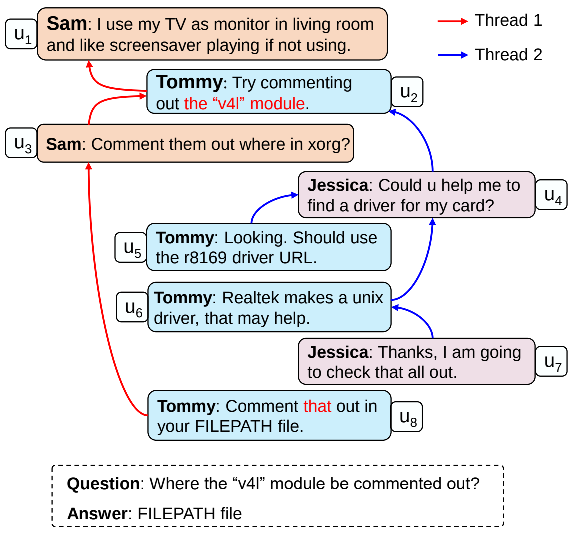

Thread 1
Thread 2
u1 Sam: I use my TV as monitor in living room and like screensaver playing if not using.
u2 Tommy: Try commenting out the "v4l" module.
u3 Sam: Comment them out where in xorg?
u4 Jessica: Could u help me to find a driver for my card?
u5 Tommy: Looking. Should use the r8169 driver URL.
u6 Tommy: Realtek makes a unix driver, that may help.
u7 Jessica: Thanks, I am going to check that all out.
u8 Tommy: Comment that out in your FILEPATH file.
Question: Where the "v4l" module be commented out?
Answer: FILEPATH file

Figure 1: An example to show that utterance dependency should be modeled to address machine reading comprehension in multi-party conversations. Two threads occur concurrently in this conversation, leading to incoherence between consecutive utterances. The word "that" in red points to the phrase "the v4l module" is several utterances away from it.

This suggests that it could be beneficial to take into account structural information when processing multi-party conversations. One very intuitive idea is to encode utterance dependencies produced by heuristic rules (Hu et al., 2019) or utterance dependency parsers (Shi and Huang, 2019).

To this end, modeling utterance dependency with Transformers (Vaswani et al., 2017) appears to be quite appealing. Two common implementations are hierarchical and utterance masking methods. Hierarchical methods (Hu et al., 2019; Xiao et al., 2020; Shen et al., 2021) firstly encode each utterance separately and then feed the utterance representations into a graph neural network built according to utterance dependencies. Nevertheless, these methods are restricted to tasks that demand only utterance-level representations and may fall short in dealing with tasks such as reading comprehension that require token representations. On the other hand, utterance masking methods (Nguyen et al., 2019; Liu et al., 2021) directly mask those self-attention weights between utterances that have no dependency relation. However, our preliminary empirical evidence (Section 5.3) shows that masking methods are insensitive to different parsers and may not really capture the dependency as desired.

In this paper, we propose a simple framework that aims to endow Transformers with the generic ability to model utterance dependency. To begin with, we first acquire the utterance dependencies in a multi-party conversation with an off-the-shelf utterance dependency parser (Shi and Huang, 2019). To encode the utterance dependencies, we then propose *absolute dependency encoding* that directly encodes the absolute position of each utterance in the dependency tree and *relative dependency encoding* that encodes the relative dependency distance of two utterances. The encoded dependency information is integrated into Transformers by modifying the computation of self-attention. Without adding a new module, this framework can be easily applied to a wide range of Transformer-based language models and multi-party conversation tasks. Besides, to enhance the understanding of utterance dependency, we further introduce a span distance prediction pre-training task for the proposed framework. It serves as a complement to previous language modeling tasks to make the model better comprehend utterance dependency.

We implement the new framework in pre-trained language models RoBERTa (Liu et al., 2019) and BART (Lewis et al., 2020) and evaluate them on four multi-party conversation benchmark tasks, including utterance-level emotion detection, relation extraction, machine reading comprehension, and abstractive summarization. Experimental results show that this framework successfully boosts the performance of the two models on all the tasks. Besides, several conclusions can be drawn. First, while the two encoding implementations can both learn utterance dependency, the effect of relative dependency encoding is more significant, serving as an inductive bias for conversation modeling. Second, for the new pre-training task, updating only parameters related to utterance dependency while freezing the rest is more effective than updating the whole set of model parameters. Third, with explicit utterance dependency modeling, language models such as RoBERTa cause fewer inference errors than with only implicit patterns within utterances.

2 Related Work

In this section, we briefly review related work on utterance dependency in multi-party conversations and its combination with Transformers.

2.1 Utterance Dependency

Utterance dependency in multi-party conversations, which represents the linguistic relations from one utterance to another, has drawn increasing attention from the community in recent years. Asher et al. (2016) released the first annotated dataset, STAC, for utterance dependency parsing in multi-party conversations. Li et al. (2020) proposed Molweni which consists of two subtasks, namely utterance dependency parsing and machine reading comprehension. In the meantime, several utterance dependency parsers have been developed. Afantenos et al. (2015) proposed a two-stage method based on Maximum Spanning Tree (MST). On top of Afantenos et al. (2015), Perret et al. (2016) replaced the MST with Integer Linear Programming (ILP). Shi and Huang (2019) proposed a deep sequential model for dependency parsing in multi-party dialogues, which is the current state of the art for this task.

Note that utterance dependency parsing is also referred to as discourse parsing in some literature. We avoid using this term to distinguish utterance dependency parsing from document discourse parsing (Braud et al., 2017), a task to parse constituency relations rather than dependency relations.

2.2 Utterance Dependency in Transformers

Considerable efforts have been devoted to combining utterance dependency of certain types with Transformers (Vaswani et al., 2017). GSN (Hu et al., 2019) firstly encodes each utterance with an utterance encoder and then feeds the utterance representations into an utterance-level graph network constructed based on speaker relations. DAG-ERC (Shen et al., 2021) is similar to GSN but builds the graph network based on manual rules. Xiao et al. (2020) proposed a hierarchical Transformer structure, in which the attention weights of a document-level Transformer are synthesized by discourse information. Jia et al. (2020) proposed to extract several threads from the dialogue history according to dependency relations and encode each thread with an individual Transformer. Recently, Liu et al. (2021) directly masked the attention weights between irrelative utterances in Transformers while encoding the whole conversation.

Unlike the above efforts, we intend to develop a simple and generic framework to model utterance dependency in multi-party conversations without any customized structure for various multi-party conversation tasks. In particular, we explore two encoding schemes of utterance dependency and implement them in Transformer-based models by modifying the computation of self-attention. This generic nature makes it applicable to a wide range of pre-trained language models. Moreover, we introduce a span distance prediction pre-training task for the framework to enhance its understanding of utterance dependency, which serves as a complement to canonical language modeling objectives.

3 Methodology

In this section, we first formulate the concept of utterance dependency and then present our two approaches to encoding utterance dependency and the pre-training task of span distance prediction.

3.1 Utterance Dependency

A multi-party conversation can be formulated as a sequence of utterances $\mathcal{U} = \{u_0, u_1, ..., u_{N-1}\}$, where N is the number of utterances in this conversation. Each utterance u_i comprises a sequence of tokens uttered by a speaker, while the set of speakers for the utterances is denoted as $\mathcal{S}$. These utterances are concatenated into a sequence as input to our model, where a function $I(i)$ is defined to map the ith token in the input to its utterance.

As illustrated in Figure 1, utterance dependency provides structural information of some kind in multi-party conversations and can be defined by linguistic relations from one utterance to another. For example, these relations may include *comment*, *elaboration*, *contrast*, and *explanation* (Asher et al., 2016). Nevertheless, for the sake of simplicity, we only consider the connectivity of two utterances in this work while ignoring their relation type. Formally, the utterance dependency for a conversation is represented by a set $\mathcal{E}$ of directed relations in the form of $u_i \rightarrow u_j$. To extract these relations, an utterance dependency parser is applied:

$$\mathcal{E} = \text{Parser}(\{u_0, u_1, ..., u_{N-1}\}, \mathcal{S}). \quad (1)$$

The utterance dependency parser forces each utterance to point to exactly one of its preceding utterances in the same conversation. Correspondingly, the parsing result of the conversation naturally forms a tree structure with u_0 as its root node.

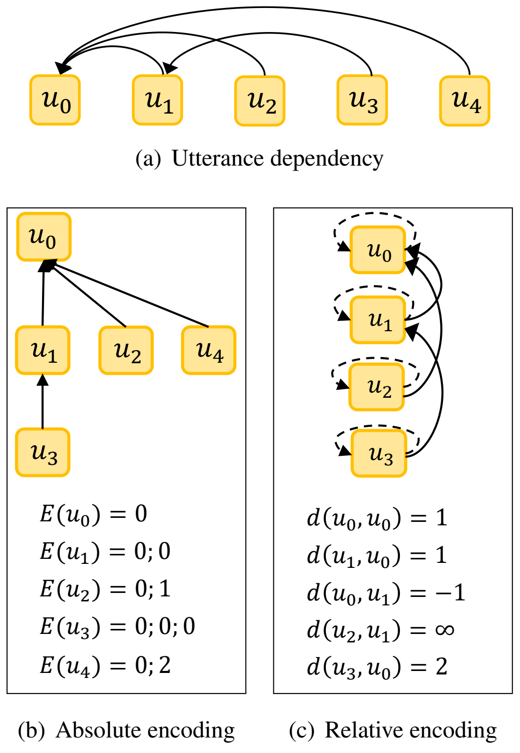


Figure 2: Demonstration of (a) utterance dependency, (b) absolute dependency encoding, and (c) relative dependency encoding. While (b) shows how to convert utterance dependency to a tree and then implement absolute utterance dependency encoding, (c) shows a subgraph of (a) and part of the relative utterance distances.

3.2 Utterance Dependency Encoding

Previous works (Yang et al., 2019; Raffel et al., 2019; Chen et al., 2021) have shown that using a positional bias term in each Transformer self-attention computation to represent the position information can improve model performance. We adopt a similar idea to encode the utterance dependency in multi-party conversations at the token level. Specifically, for each attention head in the Transformer layers, the attention score between the ith and jth tokens in the input is computed as:

$$s_{ij} = \frac{x_i W_q W_k^\top x_j^\top + a_{ij}}{\sqrt{d}}, \quad (2)$$

where $x_i W_q W_k^\top x_j^\top$ is the original attention term in Transformer (Vaswani et al., 2017), x_i and x_j are token representations, and a_{ij} is our utterance dependency bias term to represent the degree of utterance dependency for the ith and jth tokens. In the following, we investigate two approaches of absolute dependency encoding and relative dependency encoding to compute this bias term.

3.2.1 Absolute Dependency Encoding

Absolute utterance dependency encoding is based on the absolute position of an utterance in the dependency tree. For utterance u_i, its absolute dependency encoding is defined as the concatenation of its parent's absolute dependency encoding and its ranking among its siblings in temporal order:

$$E(u_i) = \begin{cases} 0, & i = 0 \\ E(P(u_i)); R(u_i), & \text{Otherwise} \end{cases} \tag{3}$$

where $P(\cdot)$ denotes the parent utterance of an utterance, $E(\cdot)$ is the absolute dependency encoding of an utterance, and $R(\cdot)$ is the ranking of an utterance among its siblings. The symbol ";" is used to represent the concatenation of two encodings. In Figure 2(b), we present an example to show how to encode absolute utterance dependency.

We then derive the absolute dependency representation of a token via a trainable embedding layer:

$$p_i = \text{Embed}_a(E(u_{I(i)})). \tag{4}$$

Finally, the attention score with absolute utterance dependency encoding can be re-computed as:

$$s^a_{ij} = \frac{x_i W_q W_k^\top x_j^\top + p_i W_q^a W_k^{a\top} p_j^\top}{\sqrt{d}}, \tag{5}$$

where W_q^a and W_k^a are trainable weight matrices.

3.2.2 Relative Dependency Encoding

We take the distance between two utterances in the dependency tree to implement the relative utterance dependency encoding. Specifically, the distance from u_i to u_j, denoted as $d(u_i, u_j)$, is determined by the length of the path from u_i to u_j in the tree. We also define the reverse distance from u_j to u_i as the negative length of this path. If such a path does not exist, both $d(u_i, u_j)$ and $d(u_j, u_i)$ are set to ∞. Especially, we define $d(u_i, u_i) = 1$ for any i, assuming that each utterance is self-dependent. Figure 2(c) presents an example of how to compute the relative utterance distance. Moreover, if two utterances are located far away from each other in the tree, their interaction is weak and can be considered as no dependency relation for simplicity. To this end, we clip the utterance distance to get the final relative utterance dependency encoding:

$$\hat{d}(u_i, u_j) = \begin{cases} d(u_i, u_j), & |d(u_i, u_j)| \le \tau \\ \infty, & \text{otherwise} \end{cases} \tag{6}$$

where τ is the threshold used to clip the dependency distance, which is a tunable hyperparameter.

The relative dependency representation between tokens u_i and u_j is obtained by feeding the relative dependency encoding of their corresponding utterances through a trainable embedding layer:

$$r_{ij} = \text{Embed}_r(\hat{d}(u_{I(i)}, u_{I(j)})). \tag{7}$$

The attention score in Equation (2) with relative utterance dependency encoding is re-computed as:

$$s^r_{ij} = \frac{x_i W_q W_k^\top x_j^\top + W^{r\top} r_{ij}}{\sqrt{d}}, \tag{8}$$

where W^r is a trainable weight matrix.

3.3 Pre-training for Dependency Encoding

Following common practice, it would be desirable to further pre-train the model on a dependency-related task to better initialize its parameters for utterance dependency encoding. Xu et al. (2021) proposed to pre-train a language model by predicting the syntax distance of two tokens to enhance its understanding of syntax information. In our scenario of utterance dependency, however, the dependency distance between two tokens can be easily inferred from the dependency distance of neighboring tokens, making this task trivial to train.

To address this, we propose to pre-train the model via a span distance prediction task. Specifically, for the input sequence we first randomly sample several pairs of text spans whose lengths follow a Poisson distribution. Then, for each pair of spans, the model is trained to predict the dependency distance between any pair of tokens from the two spans based on the output hidden states of the last encoder layer. Since the distances between tokens within a pair of spans are unknown, they cannot easily be inferred from surrounding tokens. As a result, the model is expected to understand utterance dependency better after the pre-training.

Language models such as RoBERTa (Liu et al., 2019) have been well pre-trained on large-scale corpora, which are impossible to find for multi-party conversations. To avoid forgetting the original pre-trained parameters of these models catastrophically, we opt to update only the parameters related to utterance dependency encoding, i.e., W_q^a, W_k^a, W^r, and the embedding layers of utterance dependency, while conducting our pre-training. Besides, to enhance the association of utterance dependency with language, we combine the span distance prediction objective and the original language modeling objective(s) while optimizing these language models.[1]

[1] Details of implementation are given in Appendix A.

4 Experimental Setup

We evaluate the proposed framework on four multi-party conversation benchmarks. In this section, we first introduce the experimental setup.

4.1 Dependency Parser

We employ the state-of-the-art utterance dependency parser Deep Sequential (Shi and Huang, 2019) to parse a multi-party conversation. This parser makes a sequential scan of utterances to predict the connections and relation types between utterances and uses them to build a discourse structure incrementally. To train this parser, we combine the two datasets of STAC (Asher et al., 2016) and Molweni-DP (Li et al., 2020) into a larger dataset.

4.2 Conversation Benchmarks

Our conversation benchmarks include:[2]

MELD (Poria et al., 2019) is a benchmark for utterance-level emotion classification. The evaluation metric for this benchmark is F1 score.

DialogRE (Yu et al., 2020) is a benchmark for relation extraction (RE), which predicts the relation type between two arguments. The evaluation metrics are precision (P), recall (R), and F1 score.

Molweni-MRC (Li et al., 2020) is a benchmark for reading comprehension (MRC), which identifies the answer span for a question. The evaluation metrics are exactly-match score (EM) and F1 score.

SAMSum (Gliwa et al., 2019) is a benchmark for abstractive summarization in multi-party conversations. The evaluation metrics for this benchmark are Rouge-1, Rouge-2, and Rouge-L.

4.3 Pre-trained Language Models

For the discriminative tasks of ERC, RE and MRC, we employ RoBERTA-base (Liu et al., 2019) as the base model. For the generative task of summarization, we employ BART-base (Lewis et al., 2020). We first initialize the model parameters with weights released by Huggingface[3] and then conduct our pre-training and fine-tuning on downstream tasks. The corpus for our pre-training is the combination of the four conversation benchmarks, the STAC and Molweni-DP corpora, and an open-source corpus (Choi et al., 2020) collected from the TV show *Friends*. The consolidated corpus contains 41,227 conversations and 442,539 utterances.

[2]Details of the four datasets are given in Appendix B.

[3]RoBERTa: https://huggingface.co/roberta-base, BART: https://huggingface.co/facebook/bart-base

4.4 Baseline Methods

We employ several strong baselines for comparison, including some combining utterance dependencies:

RoBERTa (Liu et al., 2019) and **BART** (Lewis et al., 2020) are the two base models used to implement our framework without utterance dependency.

RoBERTa$_{hr}$ first encodes each utterance with RoBERTa and then feeds utterance representations into a directed acyclic graph network (Thost and Chen, 2021) constructed in the same way as our work. This hierarchical structure does not apply to MRC and summarization.

RoBERTa$_{mask}$**/BART**$_{mask}$ masks the self-attention weight of two tokens whose dependency distance exceeds a threshold τ as defined in Section 3.2.2.

RoBERTa$_{abs}$**/RoBERTa**$_{rel}$**/BART**$_{abs}$**/BART**$_{rel}$ is RoBERTa/BART with our absolute utterance dependency encoding or relative utterance dependency encoding. They are initialized from the original RoBERTa/BART and directly fine-tuned on downstream tasks without further pre-training.

RoBERTa$_{abs}$**+sdp / RoBERTa**$_{rel}$**+sdp / BART**$_{abs}$**+sdp / BART**$_{rel}$**+sdp** is our proposed models further pre-trained via the span distance prediction task as described in Section 3.3.

5 Results and Analysis

We report the experimental results on the four benchmark tasks and conduct in-depth analyses.

5.1 Results on ERC, RE and MRC

The experimental results on MELD, DialogRE and Molweni-MRC are reported in Table 1. We observe that RoBERTa$_{abs}$ and RoBERTa$_{rel}$, which encode utterance dependency, outperform RoBERTa and other existing baseline models considerably, confirming the effectiveness of the two encoding strategies. After pre-training with span distance prediction, the two models see further considerable improvement on these benchmarks, also verifying the effectiveness of this pre-training task. Besides, the results show that relative utterance dependency encoding generally outperforms absolute utterance dependency encoding.

5.2 Results on Summarization

The results of conversation summarization on SAMSum are reported in Table 2. Note that we only add utterance dependency encoding in the

Method	MELD	DialogRE			Molweni-MRC	
	F1 (σ)	P (σ)	R (σ)	F1 (σ)	EM (σ)	F1 (σ)
RoBERTa	61.01 (0.31)	62.50 (0.59)	61.50 (0.59)	61.99 (0.30)	52.47 (0.65)	66.45 (0.67)
RoBERTa$_{hr}$	61.44 (0.08)	50.53 (2.75)	41.57 (3.14)	45.43 (1.12)	-	-
RoBERTa$_{mask}$	61.45 (0.56)	60.14 (0.39)	51.19 (0.47)	55.31 (0.44)	53.20 (0.76)	67.08 (0.61)
RoBERTa$_{abs}$	**62.38** (0.37)	**64.56** (1.49)	62.48 (1.25)	63.47 (0.06)	53.86 (0.85)	67.93 (0.50)
RoBERTa$_{rel}$	62.21 (0.26)	64.45 (0.86)	**63.65** (0.14)	**64.04** (0.38)	**54.25** (0.28)	**68.62** (0.58)
RoBERTa$_{abs}$+sdp	62.72 (0.24)	65.47 (1.43)	62.19 (1.22)	63.77 (0.33)	53.69 (0.22)	68.25 (0.52)
RoBERTa$_{rel}$+sdp	**62.92** (0.28)	**65.99** (0.70)	**62.96** (0.48)	**64.46** (0.16)	**55.03** (0.55)	**68.95** (0.28)

Table 1: Overall results on MELD, DialogRE and Molweni-MRC, where σ represents standard deviation.

Method	Rouge-1 (σ)	Rouge-2 (σ)	Rouge-L (σ)
BART	50.15 (0.36)	25.48 (0.09)	46.00 (0.18)
BART$_{mask}$	50.74 (0.16)	25.76 (0.14)	46.40 (0.08)
BART$_{abs}$	51.24 (0.10)	26.13 (0.10)	46.85 (0.09)
BART$_{rel}$	**51.29** (0.03)	**26.14** (0.08)	**47.13** (0.10)
BART$_{abs}$ + sdp	51.50 (0.04)	26.18 (0.03)	47.04 (0.04)
BART$_{rel}$ + sdp	**51.98** (0.18)	**26.48** (0.04)	**47.45** (0.06)

Table 2: Overall results on SAMSum, where σ represents standard deviation.

Parser	Result on MRC (F1)		
	RoBERTa$_{abs}$	RoBERTa$_{rel}$	RoBERTa$_{mask}$
MST (70.4)	67.11	67.46	67.21
Sequential (82.5)	67.43	67.84	66.97
Ground truth (100)	67.93	68.62	67.08

Table 3: Experimental results on Molweni-MRC with utterance dependency generated by different parsers. Scores in brackets are the corresponding F1 score for utterance dependency parsing on the Molweni-MRC test set.

Update	Method	MELD	DialogRE	Molweni-MRC
None	None	62.21	64.04	68.62
all	mlm	62.00	63.84	67.97
	sdp	62.28	63.77	68.17
partial	mlm	62.47	63.91	68.67
	tdp	62.58	64.21	68.77
	sdp	62.92	64.46	68.95

Table 4: F1 scores for RoBERTa$_{rel}$ with different further pre-training methods on MELD, DialogRE and Molweni-MRC. The results at the first content line of the table represent RoBERTa$_{rel}$ without further pre-training.

encoder layers of BART. Nevertheless, the performance of BART is still improved with our method, which implies that this seq-to-seq Transformer model understands the structural information in the input conversations and leads to better quality for generated texts. Besides, as the results in the discriminative tasks, a similar conclusion is that the performance of relative utterance dependency encoding is better than absolute utterance dependency encoding.

After further pre-training on span distance prediction, the performance of our BART-ABS and BART-REL are also improved, indicating that span distance prediction is also effective in seq-to-seq language models.

5.3 Quality of Utterance Dependency

We analyze how utterance dependency affects the performance of our models by using different utterance dependency results. The Molweni-MRC dataset is chosen for this experiment because it has ground truth utterance dependencies for each conversation. Three different sets of dependency results are evaluated, namely the ground truth dependency and the dependencies extracted by Deep Sequential parser and MST parser (Afantenos et al., 2015), respectively.

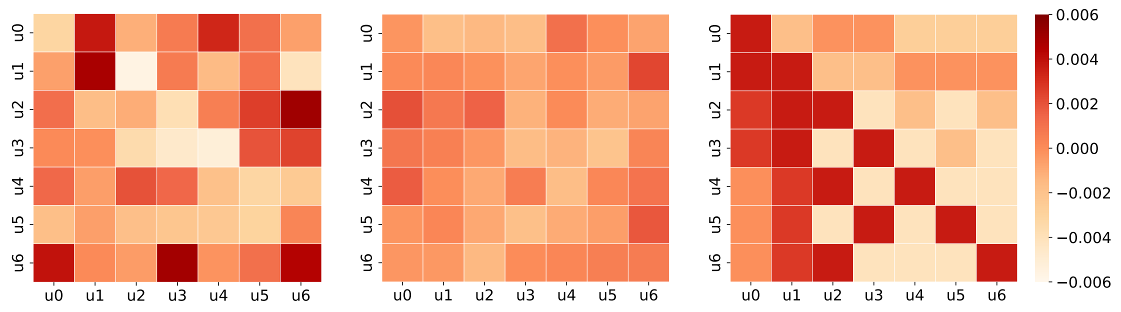
We observe that both our RoBERTa$_{abs}$ and RoBERTa$_{rel}$ suffer performance degradation due to the deterioration of quality of utterance dependency. This implies twofold. First, our proposed methods can indeed learn how to encode utterance dependency and use it to solve the MRC task. Second, good quality of utterance dependency helps solve MRC tasks.

We also conduct experiments with RoBERTa$_{mask}$. However, the performance does not show a clear trend with the quality of utterance dependency. Therefore, we suspect the masking strategies may not be sensitive enough to utterance dependency and they can hardly represent the dependency as desired.

5.4 Variants of Pre-training Methods

We conduct experiments on RoBERTa$_{rel}$ to investigate the effects of different pre-training methods on our collected corpus.[4] Our analysis is based on

[4]The results for RoBERTa$_{abs}$ are in Appendix D



(a) Segment encoding (b) Absolute dependency encoding (c) Relative dependency encoding

Figure 3: An example of the scores of bias terms added on the self-attention score computed by segment encoding, absolute dependency encoding and relative dependency encoding. Each index in the above heatmaps represents an utterance from u_0 to u_6. The utterance dependencies in the above heatmaps are $\{u_1 \rightarrow u_0, u_2 \rightarrow u_1, u_3 \rightarrow u_1, u_4 \rightarrow u_2, u_5 \rightarrow u_3, u_6 \rightarrow u_2\}$. More visualization cases are provided in the Appendix F.

Methods	MELD	DialogRE	Molweni-MRC
RoBERTa	61.01	61.99	66.45
RoBERTa$_{seg}$	61.95	62.04	67.86
RoBERTa$_{abs}$	**62.38**	63.47	67.93
RoBERTa$_{rel}$	62.21	**64.04**	**68.62**

Table 5: F1 scores on MELD, DialogRE and Molweni-MRC. RoBERTa$_{seg}$ denotes the segment encoding method introduced by Chen et al. (2021).

two aspects. First, we compare the strategies of updating all the model parameters (all) and updating only the parameters related to utterance dependency (partial). Second, we compare different pre-training tasks, including mask language modeling (mlm), our pre-training task (sdp), and replacing span distance prediction with token distance prediction (tdp) introduced by Xu et al. (2021). The results are shown in Table 4.

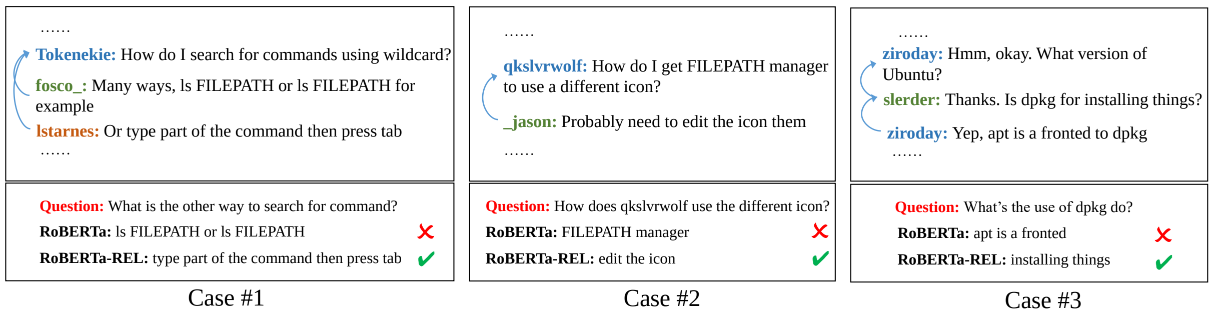
One observation from the results is that when updating all the model parameters, the performance is worse than that without pre-training. This is inconsistent with previous works that in-domain pre-training can improve model performance in downstream tasks (Sun et al., 2019; Gururangan et al., 2020). We think this is due to that we add new parameters with random initialization to the original RoBERTa. In this case, when updating the new parameters and original parameters together, the model converges to a point that is neither generalized in language representation nor fitted with utterance dependency.

If only updating the parameters related to utterance dependency, all the pre-training tasks can lead to a performance gain on RoBERTa$_{rel}$. This indicates that updating only the newly added parameters is potentially better than updating all model parameters. Besides, the performance of the three pre-training tasks generally satisfies that mlm $\leq$ tdp $\leq$ sdp, which means that designing an additional loss focusing on utterance dependency is helpful for initializing parameters related to utterance dependency encoding, while our span distance prediction is more effective than token distance prediction.

5.5 Does Utterance Dependency Encoding Only Segment the Conversations?

Since our proposed utterance dependency encoding assigns the same absolute encoding for tokens within the same utterance or the same relative encoding for token pairs within the same utterance pair, we can consider that it also plays a certain role in helping Transformers to learn how to segment the input conversation. Therefore, we want to figure out whether the performance improvement of our method truly stems from the utterance dependency or it is just because it can segment the conversation. We carry out experiments on the three discriminative tasks to compare utterance dependency encoding with the segment encoding method introduced by Chen et al. (2021). To make a fair comparison, the parameters of all compared models are initialized from the original RoBERTa-base and directly fine-tuned on the downstream tasks, without further pre-training. The results are reported in Table 5.

We note that the segment encoding method (RoBERTa$_{seg}$) can also boost the performance of RoBERTa. However, our proposed RoBERTa$_{abs}$ and RoBERTa$_{rel}$ outperform RoBERTa$_{seg}$. This indicates that the Transformer-based model equipped with our utterance dependency encoding can obtain a more comprehensive insight into conversations


……
Tokenekie: How do I search for commands using wildcard?
fosco_: Many ways, ls FILEPATH or ls FILEPATH for example
lstarnes: Or type part of the command then press tab
……

Question: What is the other way to search for command?
RoBERTa: ls FILEPATH or ls FILEPATH ✗
RoBERTa-REL: type part of the command then press tab ✓

Case #1

……
qkslvrwolf: How do I get FILEPATH manager to use a different icon?
_jason: Probably need to edit the icon them
……

Question: How does qkslvrwolf use the different icon?
RoBERTa: FILEPATH manager ✗
RoBERTa-REL: edit the icon ✓

Case #2

……
ziroday: Hmm, okay. What version of Ubuntu?
slerder: Thanks. Is dpkg for installing things?
ziroday: Yep, apt is a fronted to dpkg
……

Question: What's the use of dpkg do?
RoBERTa: apt is a fronted ✗
RoBERTa-REL: installing things ✓

Case #3

Figure 4: Three test cases in Molweni-MRC, illustrating the common errors made by RoBERTa when making reasoning and how our proposed method can rectify these errors by using utterance dependency.

rather than merely understanding the segment information. In Figure 3, we show an example of the scores of the bias term computed by different encoding methods. The figures show that our relative utterance dependency encoding imposes a direct and accurate inductive bias about utterance dependency on the model, which means that the attention weight between two tokens should also be decided by the dependency between the corresponding utterances. In contrast, we can hardly tell what the model has learned by segment encoding and absolute utterance dependency encoding. This phenomenon confirms our intuition that relative utterance dependency encoding does learn the utterance dependency, which explains why its outperforms other methods including absolute utterance dependency encoding.

5.6 Case Study

In Figure 4, we present some cases to show more clearly why the model with our proposed utterance dependency encoding can surpass the original RoBERTa. We find that RoBERTa tends to make reasoning only based on language patterns in conversations. For example, the question in Case #1 is "What is the other way ...". RoBERTa finds the answer with the pattern of "... or ..." in the second utterance. In Case #2 the question is "How does ... use the different icon", when RoBERTa finds the answer based on the pattern "get ... to ..." in the first utterance, without considering the true meaning of this utterance. The same situation happens in Case #3 where RoBERTa finds an answer based on the pattern of "... to ..." for the question with "What is the use of ...". On the contrary, in the given three cases, our RoBERTa$_{rel}$ makes reasoning based on the question-answer dependency between utterances, and yields correct results. This indicates that our proposed methods can rectify the erroneous reasoning behavior of RoBERTa by considering utterance dependency. These cases also show that utterance dependency is essential to solve some tasks on multi-party conversations.

6 Conclusion

In this paper, we designed a simple and generic framework to enhance the modeling of utterance dependency in Transformers to facilitate the understanding of conversations. Particularly, we propose absolute utterance encoding and relative utterance dependency encoding and combine them in Transformers by modifying the computation of self-attention. We also propose a pre-training task of span distance prediction to endow an utterance dependency-aware initialization. Experiments on four multi-party conversation benchmarks show that this framework is able to boost the generic performance of Transformer-based language models. With the results of further analyses, we show that utterance dependency is helpful for multi-party conversations, and our proposed methods can indeed learn to model utterance dependency and use it to solve tasks. We also discover a partial updating strategy which is more helpful than updating all model parameters in our further pre-training. We consider that we still have not exploited the full potential of utterance dependency, in future works, we will explore the way to use the relations types of utterance dependency and combine utterance dependency with other essential factors in conversation.

References

Stergos Afantenos, Eric Kow, Nicholas Asher, and Jérémy Perret. 2015. Discourse parsing for multi-party chat dialogues. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 928–937.

Nicholas Asher, Julie Hunter, Mathieu Morey, Benamara Farah, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the stac corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2721–2727.

Chloé Braud, Maximin Coavoux, and Anders Søgaard. 2017. Cross-lingual rst discourse parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 292–304.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.

Pu-Chin Chen, Henry Tsai, Srinadh Bhojanapalli, Hyung Won Chung, Yin-Wen Chang, and Chun-Sung Ferng. 2021. Demystifying the better performance of position encoding variants for transformer. *arXiv preprint arXiv:2104.08698*.

Jinho Choi, George Mihaila, and arianakc. 2020. emorynlp/character-mining. Website. `https://github.com/emorynlp/character-mining`.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *EMNLP-IJCNLP 2019*, pages 70–79.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. 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-19*, pages 5010–5016. International Joint Conferences on Artificial Intelligence Organization.

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.

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.

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.

Longxiang Liu, Zhuosheng Zhang, Hai Zhao, Xi Zhou, and Xiang Zhou. 2021. Filling the gap of utterance-aware and speaker-aware representation for multi-turn dialogue. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, pages 13406–13414.

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 Vlad 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.

Xuan-Phi Nguyen, Shafiq Joty, Steven Hoi, and Richard Socher. 2019. Tree-structured attention with hierarchical accumulation. In *International Conference on Learning Representations*.

Jérémy Perret, Stergos Afantenos, Nicholas Asher, and Mathieu Morey. 2016. Integer linear programming for discourse parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 99–109.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. Meld: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

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

Zhouxing Shi and Minlie Huang. 2019. A deep sequential model for discourse parsing on multi-party dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7007–7014.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Veronika Thost and Jie Chen. 2021. Directed acyclic graph neural networks. In *International Conference on Learning Representations*.

David Traum. 2003. Issues in multiparty dialogues. In *Workshop on Agent Communication Languages*, pages 201–211. Springer.

David C Uthus and David W Aha. 2013. The ubuntu chat corpus for multiparticipant chat analysis. In *2013 AAAI Spring Symposium Series*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wen Xiao, Patrick Huber, and Giuseppe Carenini. 2020. Do we really need that many parameters in transformer for extractive summarization? discourse can help ! In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 124–134, Online. Association for Computational Linguistics.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 2021. Syntax-enhanced pre-trained model. 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 5412–5422, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4927–4940.

A Implementation details for span distance prediction

When implementing our span distance prediction, for an input sequence with length l, we randomly sample $\lfloor l/10 \rfloor$ pairs of spans, the lengths of which are sampled in a Poisson distribution with $\lambda = 10$. We denote the set of sampled pair of spans as $\mathcal{S}$, each pair of spans as (s_1, s_2), the corresponding indices of the tokens in the spans as $s_{11}, s_{12}, ...s_{1l_1}$ and $s_{21}, s_{22}, ...s_{1l_2}$. The loss function of span distance prediction is computed as:

$$\mathcal{L}_{sdp} = -\sum_{(s_1,s_2)\in\mathcal{S}} \sum_{i=s_{11}}^{s_{1l_1}} \sum_{j=s_2}^{s_{2l_2}} \hat{d_{ij}} \log(d_{ij}) \quad (9)$$

Where d_{ij} is the predicted distance between ith and jth tokens in the input text and $\hat{d_{ij}}$ is the ground-truth distance.

The final loss function for our further pre-training task is the sum of $\mathcal{L}_{sdp}$ and the loss function for language modeling:

$$\mathcal{L} = \mathcal{L}_{sdp} + \mathcal{L}_{lm} \quad (10)$$

For pre-training RoBERTa$_{rel}$, since the input relative utterance dependency encodings are also relative distances between tokens, to avoid leaking of labels, we adopt a mask strategy for the distances within the sampled spans, specifically, we randomly mask 80% of the distances, and 10% are replaced by a random distance, and 10% are remained as the same.

We carried out the further pre-training tasks on 4 NVIDIA GeForce RTX 3090 GPUs. The batch size is set 16, learning rate is set to 5e-5. We further pretrain the models on our collected corpus for 20000 steps, with the first 4000 steps as warm-up steps. The best checkpoints for the down-stream tasks are selected among the checkpoints at 5000, 10000, 15000 and 20000 steps, based on the performance on the validation set of the downstream tasks.

B Implementation Details for Downstream Tasks

In this section we introduce the implementation details for four downstream tasks in this paper.

B.1 Implementation Details for MELD

	Train	Dev	Test
# Conversations	1038	114	280
# Utterances	9989	1109	2610

Table 6: Statistics for MELD.

The statistics for MELD are shown in Table 6.

In MELD, models are ask to predict the emotion of each utterance. There are 7 emotion labels including *neutral*, *happiness*, *surprise*, *sadness*, *anger*, *disgust*, and *fear*. When predicting the emotion of u_t, the model input is the special token <s> followed by the concatenation of utterances from u_0 to u_t, and utterances are separated by special token </s>, namely $\{$<s> u_0 </s> $u_1 ... u_{t-1}$ </s> $u_t\}$. We take the hidden state of the <s> at the last encoder layer as the final representation of the predicted sample, and pass it to a multi-layer perceptron (MLP) to get the final prediction y. The models are trained on minimizing the cross-entropy loss:

$$\mathcal{L} = -\frac{1}{M}\sum_{i=1}^{M} \hat{y_i}\log(y_i) \tag{11}$$

Where M is the number of samples, y_i is the model predicted emotion, and $\hat{y_i}$ is the ground-truth label.

B.2 Implementation Details for DialogRE

	Train	Dev	Test
# Conversations	1073	358	357
# Utterances	14024	4685	4420
# Argument pairs	5997	1914	1862

Table 7: Statistics for DialogRE.

The statistics for DialogRE are shown in Table 7.

In DialogRE, given a conversation context $\{u_0, u_1, ..., u_{N-1}\}$ and two arguments a_1, a_2, models are asked to predict the potential relation types between a_1 and a_2. DialogRE is a multi-label classification task, there are 37 types of relations, including 36 regular relations and 1 relation as no relation. The model input for DialogRE is the special token <s> followed by the concatenation of utterances from u_0 to u_{N-1} and two arguments a_1, a_2, the utterances and arguments are separated by special token </s>, namely $\{$<s> u_0 </s> $u_1 ... u_{N-1}$</s> a_1 </s> $a_2\}$. It is worth noting that since the two arguments are in the model input, we need to design their utterance dependency encoding. The method is quite straightforward, specifically, for absolute utterance dependency encoding, we give the two arguments a special absolute utterance dependency encoding that is different from the utterances, and for relative utterance dependency encoding, we set the relative dependency distance between arguments and all utterances as 1.

We take the hidden state of the <s> at the last encoder layer as the final representation of the predicted sample, and pass it to 37 separated MLPs, each MLP will predict a $y^c \in \{0, 1\}$, which represents whether there is relation type c between the two arguments or not. The models are trained on minimizing the cross-entropy loss:

$$\mathcal{L} = -\frac{1}{M}\sum_{i=1}^{M}\sum_{c=1}^{37} \hat{y_i^c}\log(y_i^c) \tag{12}$$

Where M is the number of samples, y_i^c is the model predicted label for relation type c, and $\hat{y_i^c}$ is the ground-truth label for relation type c.

B.3 Implementation Details for Molweni-MRC

	Train	Dev	Test
# Conversations	8771	883	100
# Utterances	77374	7823	845
# Questions	24682	2513	2871

Table 8: Statistics for Molweni-MRC.

The statistics for Molweni-MRC are shown in Table 8.

In Molweni-MRC, given the conversation context $\{u_0, u_1, ..., u_{N-1}\}$ and a question q, the models are asked to predict the start and end positions of the answer span in the conversation context. The model input is the question concatenated with the conversation context, namely $\{$<s> q </s> u_0...</s> $u_{N-1}\}$. Similar to DialogRE, for absolute utterance dependency encoding, we give the question a special absolute utterance dependency encoding that is different from the utterances, and for relative utterance dependency encoding, we set the relative dependency distance between question and all utterances as 1.

The models are trained on minimizing the cross-entropy loss:

$$\mathcal{L} = -\frac{1}{M}\sum_{i=1}^{M}(\hat{y_i^s}\log(y_i^s) + \hat{y_i^e}\log(y_i^e)) \tag{13}$$

Where M is the number of samples, y_i^s and y_i^e are the model predictions for start position and end position, respectively, and $\hat{y_i^s}$ and $\hat{y_i^e}$ are the ground-truth labels.

B.4 Implementation Details for SAMSum

The statistics for SAMSum are shown in Table 9.

The model input for SAMSum is the concatenation of utterances, with the utterances separated by </s>, namely

	Train	Dev	Test
# Conversations	14731	818	819
# Utterances	164505	8860	9212

Table 9: Statistics for SAMSum.

$\{$`<s>` u_0 `</s>` u_1... u_{N-2} `</s>` $u_{N-1}\}$. The models are trained on minimizing the negative log-likelihood:

$$\mathcal{L} = -\frac{1}{M}\sum_{i=1}^{M}\sum_{t=1}^{l} p(y_{i,t}|y_{i,<t}) \tag{14}$$

Where M is the number of samples, $y_{i,t}$ is the tth token of the generated summary.

B.5 Overall implementation details

We train the models on downstream tasks with 4 NVIDIA GeForce RTX 2080 Ti GPUs. The hyper-parameters that can be tuned for model selection include learning rate, batch size, and the threshold τ described in Section 3.2.2. The best model is selected by the hold-out validation on the validation set. All the experimental results reported in this paper are achieved by carrying out 3 random tests and averaging their results.

C Implementation Details for absolute dependency encoding

When implementing absolute dependency encoding, if we cover all the possible absolute dependency encodings, the size of the memory demanded for storing the absolute dependency encodings and the embedding layers will experience exponential increment as the number of utterances increases. To tackle this problem, we build a vocabulary for the absolute dependency encodings appear in the training set and the validation set for each downstream task. When evaluating in the test set, if an utterance's absolute dependency encoding can not be found in the vocabulary, it is marked as a special `UNK` encodings, otherwise it is kept as the same. This method saves memory significantly, and its coverage rate for the absolute dependency encodings on test set is acceptable. Specifically, the coverage rates for the test set of MELD, DialogRE, Molweni-MRC, SAMSum are 94.03%, 92.56%, 99.32% and 95.34%, respectively.

D RoBERTa$_{abs}$ with variants of further pre-training tasks

In this section we show the results of RoBERTa$_{abs}$ with further pre-training tasks introduced in Section 5.4. The results are reported in Table 10. We can find that the reported results are consistent with the discoveries made in Section 5.4.

Update	Method	MELD	DialogRE	Molweni-MRC
None	None	62.38	63.47	67.93
all	mlm	62.12	63.35	67.38
	sdp	62.32	63.43	67.32
partial	mlm	62.42	63.56	68.27
	tdp	62.59	63.67	68.01
	sdp	62.72	63.77	68.25

Table 10: F1 scores for RoBERTa$_{abs}$ with different further pre-training methods. The results at the first content line of the table represent RoBERTa$_{abs}$ without further pre-training.

E Results for comparing with language models with further language modeling

Method	MELD F1 (σ)	DialogRE F1 (σ)	Molweni-MRC F1 (σ)
RoBERTa + lm	61.70 (0.36)	60.12 (0.93)	67.03 (0.73)
RoBERTa$_{abs}$ + sdp	62.72 (0.24)	63.77 (0.33)	68.25 (0.52)
RoBERTa$_{rel}$ + sdp	**62.92** (0.28)	**64.46** (0.16)	**68.95** (0.28)

Table 11: Experimental results for ERC, RE and MRC. RoBERTa+lm represents RoBERTa further pre-trained with the mask language modeling task.

Method	Rouge-1 (σ)	Rouge-2 (σ)	Rouge-L (σ)
BART + lm	51.18 (0.18)	26.16 (0.26)	46.83 (0.23)
BART$_{abs}$ + sdp	51.50 (0.04)	26.18 (0.03)	47.04 (0.04)
BART$_{rel}$ + sdp	**51.98** (0.18)	**26.48** (0.04)	**47.45** (0.06)

Table 12: Experimental results in conversation summarization. BART+lm represents BART further pre-trained with the mask language recovering task.

We also compared our methods with the pre-trained language models further pre-trained by language modeling tasks on our collected corpus. The results are reported in Table 11 and Table 12. As shown in the results, our methods with our proposed further pre-training task can still outperform the pre-trained language models with further language modeling tasks.

F More visualization for the case in Section 5.4

We show more visualizations of different layers and heads for the case described in Section 5.4.

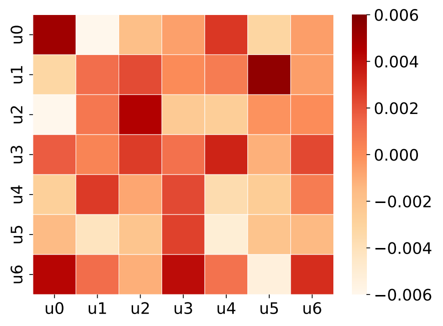


Figure 5: RoBERTa$_{seq}$ layer_0_head_0

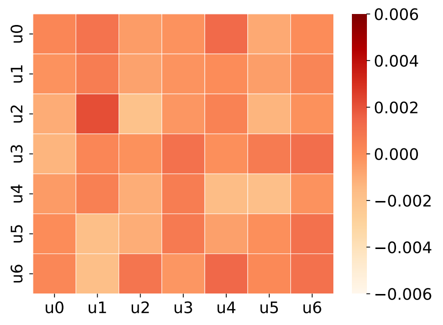


Figure 6: RoBERTa$_{abs}$ layer_0_head_0

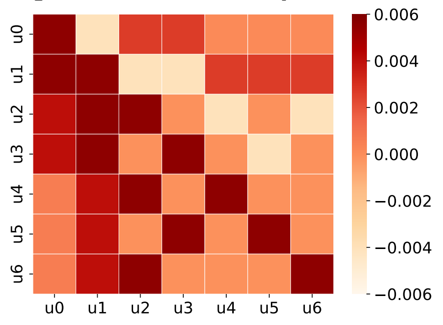


Figure 7: RoBERTa$_{rel}$ layer_0_head_0

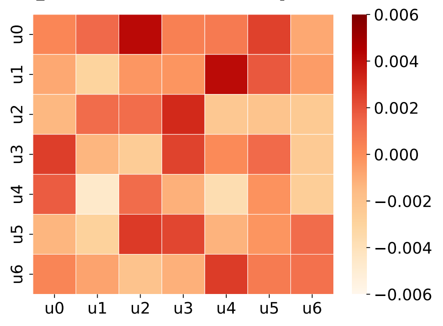


Figure 8: RoBERTa$_{seq}$ layer_5_head_4

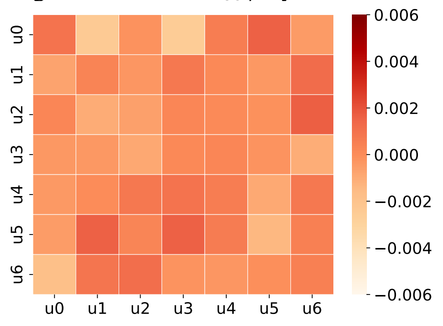


Figure 9: RoBERTa$_{abs}$ layer_5_head_4

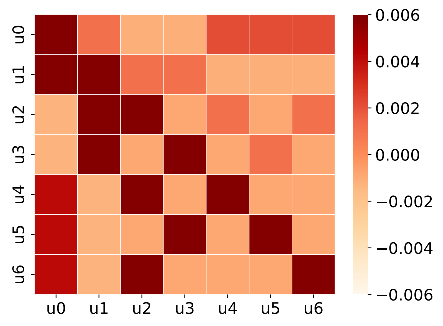


Figure 10: RoBERTa$_{rel}$ layer_5_head_4

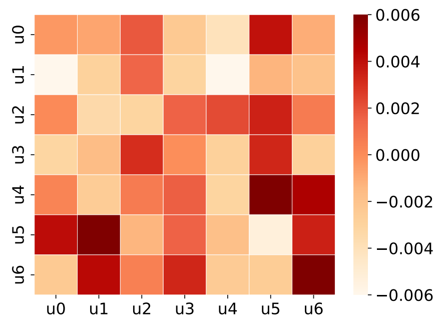


Figure 11: RoBERTa$_{seg}$ layer_11_head_3

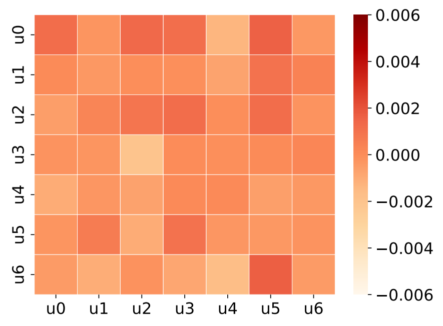


Figure 12: RoBERTa$_{abs}$ layer_11_head_3

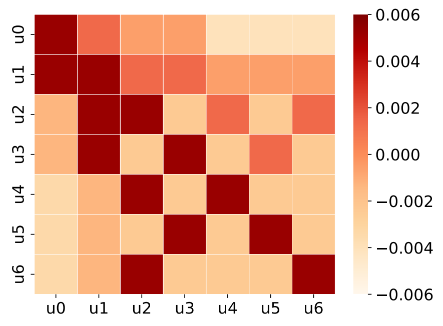


Figure 13: RoBERTa$_{rel}$ layer_11_head_3