Conversation- and Tree-Structure Losses for Dialogue Disentanglement

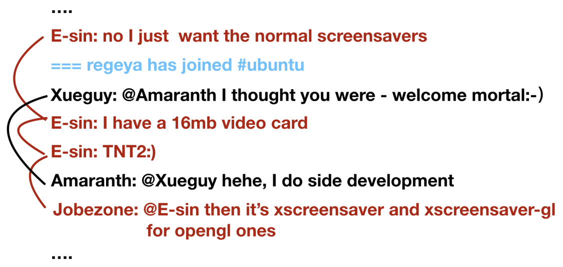
Tianda Li[1], **Jia-Chen Gu**[2], **Zhen-Hua Ling**[2], **Quan Liu**[3]
[1]Nankai University, Tianjin, China
[2]National Engineering Research Center for Speech and Language Information Processing,
University of Science and Technology of China, Hefei, China
[3]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, Hefei, China
focuslee1214@gmail.com, gujc@mail.ustc.edu.cn
zhling@ustc.edu.cn, quanliu@iflytek.com

Abstract

When multiple conversations occur simultaneously, a listener must decide which conversation each utterance is part of in order to interpret and respond to it appropriately. This task is referred as *dialogue disentanglement*. A significant drawback of previous studies on disentanglement lies in that they only focus on pair-wise relationships between utterances while neglecting the conversation structure which is important for conversation structure modeling. In this paper, we propose a hierarchical model, named Dialogue BERT (DIALBERT), which integrates the local and global semantics in the context range by using BERT to encode each message-pair and using BiLSTM to aggregate the chronological context information into the output of BERT. In order to integrate the conversation structure information into the model, two types of loss of conversation-structure loss and tree-structure loss are designed. In this way, our model can implicitly learn and leverage the conversation structures without being restricted to the lack of explicit access to such structures during the inference stage. Experimental results on two large datasets show that our method outperforms previous methods by substantial margins, achieving great performance on dialogue disentanglement.

1 Introduction

In a multi-party chat stream (Traum, 2004; Uthus and Aha, 2013; Ouchi and Tsuboi, 2016; Gu et al., 2021), messages related to different topics are entangled with each other, which makes it difficult for a new user to understand the context of the discussion in the chat room. Dialogue disentanglement (Kummerfeld et al., 2019; Gu et al., 2020b; Yu and Joty, 2020; Liu et al., 2021a,b) aims at disentangling a whole conversation into several threads from a data stream so that each thread is about a specific topic. Early research either did not release their datasets (Adams and Martell, 2008; Wang et al., 2008) or used small datasets (Elsner and Charniak, 2008; Elsner and Schudy, 2009; Wang and Oard, 2009; Elsner and Charniak, 2010, 2011; Jiang et al., 2018). Kummerfeld et al. (2019) released a new large-scale dataset that made it possible to train a more complex model and to fairly compare different models. Figure 1 shows an example of dialogue disentanglement in this dataset.



....
E-sin: no I just want the normal screensavers
=== regeya has joined #ubuntu
Xueguy: @Amaranth I thought you were - welcome mortal:-)
E-sin: I have a 16mb video card
E-sin: TNT2:)
Amaranth: @Xueguy hehe, I do side development
Jobezone: @E-sin then it's xscreensaver and xscreensaver-gl for opengl ones
....

Figure 1: An example of dialogue disentanglement. In this example, conversations marked with different colours are entangled together. This task aims to separate this chat stream by conversations.

Currently, most of the existing methods for dialogue disentanglement employ a two-step approach framework. Firstly, a model is employed to determine the relation between two messages. Then a clustering algorithm is employed to separate these messages into different conversation clusters. Following this framework, Zhu et al. (2020) proposed a BERT-based model named Masked Hierarchical Transformer (MHT), which aims at making use of the conversation structures. This method uses a mask mechanism to explicitly build connections between context messages and their corresponding ancestors in a conversation. However, the main drawback of their approach is that the designed mask is computed based on the parents' relation of each message given the whole conversation, which is only available during the training stage. In order to deal with the lack of masks during the inference stage, they construct the

Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering, pages 54 - 64
May 26, 2022 ©2022 Association for Computational Linguistics

pseudo mask label based on the predicted relations between any message-pair. However, the pseudo mask label cannot introduce reliable conversation structure information, especially when models cannot achieve a perfect prediction performance on relevant datasets.

In this work, we follow this two-step approach framework and propose a hierarchical BERT-based model, named Dialogue BERT (DIALBERT) for dialogue disentanglement. DIALBERT first use BERT (Devlin et al., 2019) to capture the matching information in each message pair. Then, a context-level BiLSTM is employed to aggregate and incorporate the context information. The semantics similarity of each message pair is measured by calculating their matching scores, and the message that has the highest matching score with the target message is regarded as the parent message of it. In addition, we aim at introducing and making use of conversation structures to help DIALBERT to make decision by training DIALBERT with two extra types of loss of conversation-structure loss and tree-structure loss. In this way, the model can implicitly learn and leverage conversation structures without being restricted to the lack of explicit access to such structures during inference.

We evaluate our method on two large datasets releasaed by Kummerfeld et al. (2019) and Zhu et al. (2020) respectively. Experimental results show our proposed method outperforms previous methods in terms of various evaluation metrics.

In summary, our contributions in this paper are three-fold: (1) A hierarchical model named DIALBERT is proposed for dialogue disentanglement. (2) Two losses of conversation- and tree-structure losses are introduced to make use of the structures of the conversation history. (3) The performance of the proposed method is evaluated on two large datasets, and the ablation studies further verified the effectiveness.

2 Related Work

The research for dialogue disentanglement dates back to Aoki et al. (2003) which conducted a study of voice conversations among 8-10 people with an average of 1.76 activate conversations at any given time. In recent studies, the mainstream method for dialogue disentanglement is the two-step approach: firstly, a neural network is used to determine the relation between two messages. Then a clustering algorithm is adopted to separate messages into different conversations. In the first step, Mehri and Carenini (2017) used recurrent neural networks(RNNs) to model adjacent messages. Jiang et al. (2018) was the first work that used convolutional neural networks to estimate the conversation-level similarity between closely posted messages. Zhu et al. (2020) proposed a Masked Hierarchical Transformer based on BERT to calculate the matching score by using conversation structures. In addition to neural networks, statistical (Du et al., 2017) and linguistic features (Elsner and Charniak, 2008, 2010, 2011; Mayfield et al., 2012) have also been used in the existing research. In the clustering stage, some research proposed the clustering algorithm by using threshold such as Jiang et al. (2018). Most studies grouped two messages with the highest matching score into the same conversation. In our study, we follow this mainstream setting.

3 Problem Formulation

Given a dataset $\mathcal{D}$, $\{M^{(1)}, M^{(2)}, ..., M^{(N)}\}$ represents a list of messages and each message belongs to a specific conversation. Following the setting of previous studies (Elsner and Charniak, 2008, 2010, 2011; Mayfield et al., 2012; Jiang et al., 2018), in order to find the parent message of a target message, $T-1$ messages occurring before this target message and itself form the context message set of this target message. The target message is a word sequence that can be represented by $M^T = \{m_1^T, m_2^T, ..., m_{n_T}^T\}$, and each context message is a word sequence that can be represented by $M^i = \{m_1^i, m_2^i, ..., m_{n_i}^i\}$, where n_T and n_i are the sequence length of messages and $i \in \{1, 2, ..., T\}$. Every target message has a label $Y \in \{1, 2, ..., T\}$ indicating which message in context range is the parent message of the target message (each message has and only has one parent message). Our goal is to learn a prediction model to predict which message in $\{M^1, M^2, ..., M^T\}$ is the parent message of the target message M^T for $T \in \{1, 2, .., N\}$. Note that if the target message is the first message of a conversation, the parent of the target message is itself.

4 Methodology

4.1 DIALBERT

DIALBERT calculates the matching scores between the target message and its context messages. The overall architecture is shown in Figure 2. The

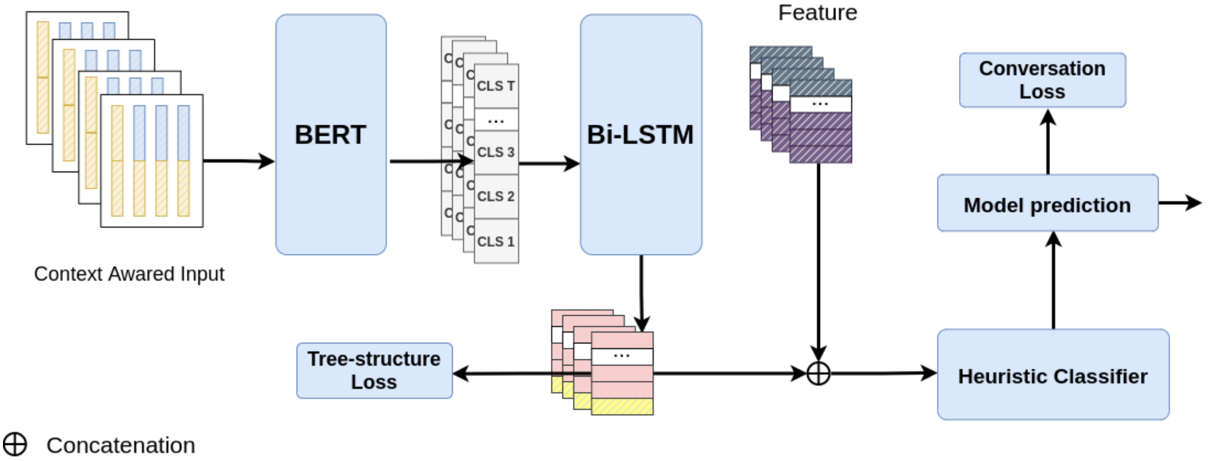


Figure 2: The overall architecture of DIALBERT. `CLS T` is the `[CLS]` hidden state of the T-th message pair. Note that the hand-craft features designed before the heuristic classifier is introduced in Kummerfeld et al. (2019). These features are not used on the Reddit dataset.

context message that has the largest matching score with the target message will be regarded as the parent message. For the second step, after we get the parent message of each target message, we group messages into different conversations based on the parental relations.

4.1.1 Context-Aware Input

In order to take context semantics in a chat into consideration, $T-1$ preceding messages of the target message are used along with the target message to form the context message set. Specifically, every context message will be concatenated with the target message to form a message pair. Then, all the message pairs will be combined together as a single input to predict the parent message of each target message. The input $\mathbf{u}_i$ can be formulated as: $\mathbf{u}_i = \left[cls, m_1^T, ..., m_{n_T}^T, sep, m_1^i, ..., m_{n_i}^i, sep\right]$, where $\mathbf{U} = \{\mathbf{u}_i\}_{i=1}^T$. $i \in [1, 2, ..., T]$ is the index of the context message. cls and sep are the start and separation tokens predefined in BERT, respectively. Note that $\mathbf{u}_T$ is composed of two target messages.

4.1.2 Context BERT Module

A strategy to consider context is to concatenate the context messages with the target message. But this strategy weakens the relationships between each context message as they are organized in chronological order in the chat stream.

In order to better consider the chronological order information of context messages, we propose a context BERT module to encode the history context by using both BERT and a BiLSTM model. Specifically, we encode input $\mathbf{U}$ by adopting BERT, and the output of the reserved cls will be used as feature vectors $\mathbf{E} = \{\mathbf{e}_i\}_{i=1}^T$. Each feature vector $\mathbf{e}_i$ contains the semantics in its corresponding message pair. In addition, we further encode the feature vectors $\mathbf{E}$ with a single layer Bi-LSTM to obtain the high-order feature vectors $\mathbf{F}$, which have captured the semantics of history context and can be represented as $\{\mathbf{f}_i\}_{i=1}^T$. The formulae of the calculation are as follows:

$$\mathbf{e}_i = \mathrm{BERT}(\mathbf{u}_i), \forall i \in [1, 2, ..., T], \quad (1)$$

$$\mathbf{f}_i = \mathrm{BiLSTM}(\mathbf{e}_i), \forall i \in [1, 2, ..., T], \quad (2)$$

$$\mathbf{m} = \mathrm{Softmax}(\mathrm{Linear}(\mathbf{F})), \quad (3)$$

where the dimension of the hidden units in a BiLSTM layer is k. $\mathbf{m} = \{m_i\}_{i=1}^T$ are matching degrees that will be used to calculate the tree-structure loss in Section 4.2.

4.1.3 Heuristic Classifier

To model the higher-order interaction between the target message and its context messages, a heuristic classifier which has proved to be effective in different studies (Yoon et al., 2018; Chen et al., 2017, 2018), is employed. Specifically, the interaction vectors $\mathbf{G} = \{\mathbf{g}_i\}_{i=1}^T$ will be fed into a single layer classifier to get matching scores, with the following formulae:

$$\mathbf{g}_i = [\mathbf{f}_i, \mathbf{f}_T, \mathbf{f}_i \circ \mathbf{f}_T, \mathbf{f}_i - \mathbf{f}_T], \quad (4)$$

$$\mathbf{p} = \mathrm{Softmax}(\mathrm{Linear}(\mathbf{tanh}(\mathbf{G}\mathbf{W}_3^{\mathrm{T}} + \mathbf{b}_3))), \quad (5)$$

where $\mathbf{W}_3 \in \mathbb{R}^{4k \times 8k}$ is weight matrix and $\mathbf{b}_3 \in \mathbb{R}^{4k}$ is the bias. $\circ$ is element-wise product and $-$ is element-wise subtraction. $\mathbf{p} = \{p_i\}_{i=1}^T$ are the matching scores, and will be used to calculate cross-entropy loss $\mathcal{L}_{CE}$ (shown below) and conversation-structure loss $\mathcal{L}_{CV}$.

$$\mathcal{L}_{CE} = -\frac{1}{T}\sum_{i=1}^{T} y_i \log{(p_i)}, \tag{6}$$

where $\{y_i\}_{i=1}^{T}$ is the one-hot embedding of golden label Y. T is the context range. The overall loss for DIALBERT model can be formalized as :

$$\mathcal{L}_{overall} = \mathcal{L}_{CE} + \alpha\mathcal{L}_{CV} + \beta\mathcal{L}_{TS}, \tag{7}$$

where α and β are hyperparameters. The conversation-structure loss $\mathcal{L}_{CV}$ and tree-structure loss $\mathcal{L}_{TS}$ will be introduced in Section 4.2. Finally, the context message with the largest matching score is regarded as the parent message of target message, and we group these two messages into the same conversation.

4.2 Conversation- and Tree-Structure Loss

In the list of messages, different conversations are entangled together, and each conversation has its own semantic coherence and cohesion. Most previous studies failed to use the structure of each conversation when the parent message of a target message in the context is determined. In order to encourage our model to find the parent message of the target message based on the context coherence of the conversation, we introduce conversation-structure loss and tree-structure loss in addition to the cross-entropy loss. In this way, our model can learn and leverage the structure of the conversation implicitly and will not suffer from a lack of conversation structure information during the inference/testing stage. Intuitively, both conversation-structure loss and tree-structure loss can encourage the model to select most relevant message as the parent message.

4.2.1 Conversation-Structure Loss

The conversation-structure loss is computed based on the matching score:

$$\mathcal{L}_{CV} = -\frac{1}{T}\sum_{i=1}^{T} y_i^c \log(p_i), \tag{8}$$

where $\{y_i^c\}_{i=1}^{T}$ are the conversation labels and each y_i^c is a binary label indicating whether the i-th context message is in the conversation same as the target message. $\{p_i\}_{i=1}^{T}$ are matching scores of message pairs. T is context range. The intention of the conversation-structure loss is to encourage the model to choose the parent message for a target message from the messages in the same conversation.

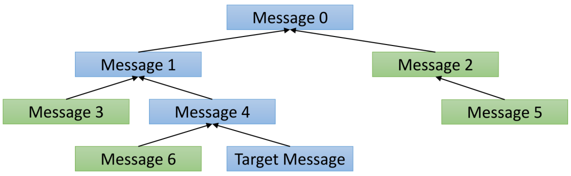


Figure 3: An example of the conversation structure. A chat stream consists of multiple these structures. Conversation-structure loss will help the model distinguish which conversation structure does target message belong to and Tree-Structure loss will help the model further distinguish ancestor messages of target message in the structure.

4.2.2 Tree-Structure Loss

In order to further make use of the structure of conversation, we propose tree-structure loss. Intuitively, in a structure of a conversation (shown in Figure 3), ancestors of the target message (i.e., message 0, message 1 and message 4) are most relevant to the target message. Because the target message can be regarded as the response to its ancestor messages or as an extension of the topic discussed in the ancestor messages, the intention of the tree-structure loss is to help the model further narrow down the candidates. The tree-structure loss encourages the model to choose the parent message for a target message from all ancestor messages in the same conversation. The tree-structure loss has two terms that are designed for ancestor nodes and other nodes, respectively. The first term of the tree-structure loss can be computed with the following formulae:

$$y_i^a = \begin{cases} \mathit{0.5} & \mathit{if} \quad d = 0, \\ \mathit{1.2\text{-}0.2{*}d} & \mathit{if} \quad 0 < d \leq 5, \\ \mathit{0.1} & \mathit{if} \quad 5 < d, \end{cases} \tag{9}$$

$$\mathcal{L}_{FirstTerm} = -\frac{1}{T}\sum_{i=1}^{T} y_i^a log(m_i), \tag{10}$$

where d is the distance between the specific context message and target message in the structure of a conversation. For example, in Figure 3, d of *message 1* and the *target message* is 2. Note that $d = 0$ is the distance for the special message pair in which the target message is paired with itself. Because our target is to find the parent message.

In order to add the penalty to the model, if non-ancestor messages in the conversation are selected as the parent of the target message, we designed three strategies for calculating the second term of the tree-structure loss: *uniform-penalty*, *penalty-by-distance*, and *penalty-by-layer-difference*. For *uniform strategy*, $y_i^b = 0.1$ if the i-th context message is not an ancestor message of the target message. For *penalty-by-distance*, the strategy is formalized as follows:

$$y_i^b = \begin{cases} \mathit{1}\text{-}\,\dfrac{d}{20} & if \quad 0 \le d < 20 \\ \mathit{0.1} & if \quad 20 \le d \end{cases}, \tag{11}$$

where d is the distance between the target message and the corresponding message in the structure of the conversation; e.g., in Figure 3, d between *message 3* and *target message* is 3. For *penalty-by-layer-difference*, the strategy can be formalized as:

$$y_i^b = \begin{cases} \mathit{1}\text{-}\,\dfrac{l_i}{10} & if \quad 0 \le l_i < 10 \\ \mathit{0.1} & if \quad 10 \le l_i \end{cases}, \tag{12}$$

$$l_i =\mid layer_{target} - layer_i \mid, \tag{13}$$

where $layer_{target}$ is the layer number of the target message in the structure of the conversation. $layer_i$ is the layer number of message i; e.g., the layer difference between *message 2* and *target message* is $\mid 4 - 2 \mid= 2$. The tree-structure loss $\mathcal{L}_{TS}$ can be formulated as:

$$\mathcal{L}_{SecondTerm} = -\frac{1}{T}\sum_{i=1}^{T} y_i^b \log(m_i), \tag{14}$$

$$\mathcal{L}_{TS} = \mathcal{L}_{FirstTerm} - \mathcal{L}_{SecondTerm}. \tag{15}$$

Note that if the i-th context message is not in the same conversation as the target message, then $y_i^a = 0$, $y_i^b = 0$.

5 Experiments

5.1 Datasets

Our proposed method was evaluated on the Ubuntu IRC dataset (Kummerfeld et al., 2019), which is manually annotated with reply-to relationship between messages. The statistics of distances between the target and its parent message is shown in Figure 4. In addition, we also evaluated our proposed method on the Reddit-large dataset proposed in Zhu et al. (2020).[1] We followed the settings in Zhu et al. (2020) to further filter the Reddit-large dataset: if a comment or the user who posted the comment is deleted, the comment itself and all its descendants are not included in the dataset. These conversations were splitted into train/validation/testing sets in a ratio of 8:1:1. The overall statistics of the two datasets are shown in Table 1 and data examples from these two datasets are shown in Table 2.

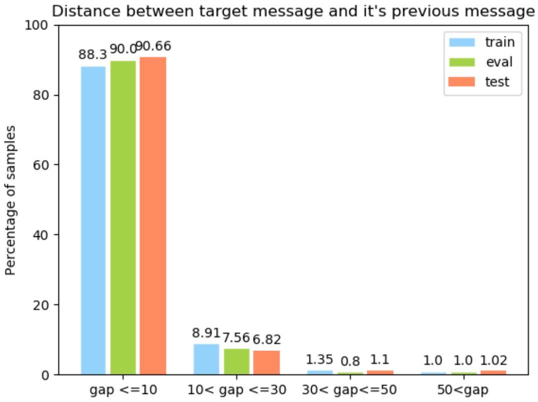


Figure 4: The percentage of distances between the target message and its parent message in the Ubuntu IRC dataset.

	Message	Conversation	Avg. Distance
Ubuntu IRC			
Train	67463	3825	6.55
Validation	2500	250	6.87
Testing	5000	280	6.16
Reddit			
Train	468679	20178	5.53
Validation	37300	2098	5.97
Testing	72933	4133	5.95

Table 1: Statistics of the Ubuntu IRC and the Reddit datasets. The last column denoted the averaged distance between a target message and its parent message.

5.2 Evaluation Metrics

For the Ubuntu IRC dataset, we follow the setting in Kummerfeld et al. (2019). The evaluation metrics used in our experiments include: the modified Variation of Information (VI) (Kummerfeld

[1] Zhu et al. (2020) only provide the comment IDs and crawling scripts. The data collected in our paper is crawled on March 23, 2020 using the provided scripts and IDs.

Parent	Index	Message
...	...	...
996	1000	[03:04] Amaranth: @cliche American
992	1001	[03:04] Xenguy: @Amaranth I thought you were – welcome mortal ;-)
1000	1002	[03:04] cliche: @ Amaranth, hahahaha
1003	1003	=== welshbyte has joined #ubuntu
997	1004	[03:04] e-sin: no i just want the normal screensavers
995	1005	[03:04] Amaranth: @benoy Do you have cygwinx installed and running?
1006	1006	[03:04] babelfishi: can anyone help me install my Netgear MA111 USB adapter?
1004	1007	[03:04] e-sin: i have a 16mb video card
1008	1008	=== regeya has joined #ubuntu
1007	1009	[03:04] e-sin: TNT2 :)
1001	1010	[03:05] Amaranth: @Xenguy hehe, i do side development
1007	1011	[03:05] jobezone: @e-sin then it's xscreensaver and xscreensave-gl for opengl ones.
1005	1012	[03:05] benoy: how do i install that? I couldn't find that in the list of things
1010	1013	[03:05] Amaranth: @Xenguy things like alacarte and easyubuntu
...	...	...
1	1	DeathisLaughing: HP forgot to print the label for this ink cartridge...that's mildly ironic...
1	2	BitJit: @ DeathisLaughing Mystery ink box! Will it fit in your printer?! no.
1	3	andrewsmith1986: @ DeathisLaughingI love this subreddit.
1	4	myfutureperfect: @ DeathisLaughing They ran out of ink. So, what?
1	5	sageDieu: @ DeathisLaughing They probably couldn't afford it
1	6	dsbaciga: @ DeathisLaughing I like that they ignore the low ink cartridge notifications just like I do.

Table 2: Data examples of the Ubuntu IRC dataset (upper) and the Reddit dataset (lower).

et al., 2019), Adjusted Rand Index (ARI), One-to-One Overlap (1-1) of the cluster (Elsner and Charniak, 2008), as well as the precision, recall, and F1 score between the cluster prediction and ground truth. Note that the precision, recall, and F1 score are calculated using the number of perfectly matching conversations, excluding conversations that have only one message (mostly system messages). We take VI as the main metric. For the Reddit dataset, we follow the setting of Zhu et al. (2020). Specifically, the graph accuracy and the conversation accuracy are adopted. The graph accuracy is used to measure the average agreement between the ground truth and predicted parent for each utterance. The conversation accuracy is used to measure the average agreement between conversation structures and predicted structures. Specifically, only if all messages in a conversation are predicted correctly, the predicted structure is regarded as correct. We take graph accuracy as the main metric.

5.3 Implementation Details

The base version of BERT was used in our experiments. The initial learning rate was set to 2e-5. The maximum sequence length was set to 100. The number of hidden unit k was 384. For the two extra losses, $\alpha = 0.15$ and $\beta = 1$ achieved the best performance. The value of α was selected from $[0.1, 0.15, 0.2]$, and that of β was selected from $[0.5, 1]$. Dropout was applied on the output layer of the *ConBERT* and heuristic classifier with a ratio of 0.1. For the IRC dataset, batch size was set to 4 and the context range T was set to 50. For the Reddit dataset, batch size was set to 3 and the context range T was set to 16. All experiments were conducted on a 24G RTX TITAN GPU. All codes were implemented in the TensorFlow framework (Abadi et al., 2016) and are published to help replicate our results. [2]

5.4 Comparison Baselines

We compare our models with those reported in Kummerfeld et al. (2019) and Zhu et al. (2020), which are shown in the Table 3. Below we list variants of our models, which are also shown in the bottom part of Table 3.

DIALBERT: Domain adaptation has shown great effectiveness to improve dialogue performance (Gu et al., 2020a; Whang et al., 2020) .In this setting, DIALBERT with adaptation [3] will be used to find parent message according to the ranking scores.

[2] https://github.com/TeddLi/Disentangle

[3] The Ubuntu forum data published in Dialog System Technology Challenges 8 (DSTC 8) - Track 2 as external data was adopted to perform domain adaptation. The input was constructed as {`[CLS], title, question, [SEP], answer, [SEP]`}. Both tasks of masked language model (MLM) and next sentence prediction (NSP) were employed during domain adaptation. Note that domain adaptation was only employed in Ubuntu IRC dataset.

	VI	ARI	1-1	F1	P	R
Linear+ feature *	88.9	-	69.5	21.8	19.3	24.9
Feedforward + feature *	91.3	-	75.6	36.2	34.6	38.0
× 10 union*	86.2	-	62.5	33.4	40.4	28.5
× 10 vote*	91.5	-	76.0	38.0	36.3	39.7
× 10 intersect*	69.3	-	26.6	32.1	67.0	21.1
Elsner(2008)*	82.1	-	51.4	15.5	12.1	21.5
Lowe(2017)*	80.6	-	53.7	8.9	10.8	7.6
Dec. Att. (dev)*	70.3	-	39.8	0.6	0.9	0.7
Dec. Att. + feature (dev)*	87.4	-	66.6	21.1	18.2	25.2
ESIM (dev)*	72.1	-	44.0	1.4	2.2	1.8
ESIM + feature (dev)*	87.7	-	65.8	22.6	18.9	28.3
BERT (dev)*	74.7	-	45.4	2.2	2.6	2.7
BERT + feature (dev)*	89.5	-	71.7	21.4	30.0	25.0
MHT (dev)*	82.1	-	59.6	8.7	12.6	10.3
MHT +feature (dev)*	89.8	-	75.4	35.8	32.7	34.2
DialBERT w/o. adapt (dev)	93.4	79.2	83.1	44.4	48.4	41.1
DialBERT (dev)	**94.1**	**81.1**	**85.6**	**48.0**	**49.5**	**46.6**
Structural Characterization (dev)	94.4	81.8	86.1	52.6	51.0	54.3
DialBERT w/o. adapt	92.5	63.5	76.5	39.8	36.4	43.8
DialBERT	92.6	69.6	78.5	44.1	42.3	46.2
DialBERT + feature	92.4	64.6	77.6	42.2	38.8	46.3
DialBERT + ensemble	93.3	75.2	-	**46.8**	**44.3**	49.6
DialBERT + cov	93.2	72.8	79.7	44.8	42.1	47.9
DialBERT + cov + uni	93.1	68.2	78.2	43.8	40.0	48.2
DialBERT + cov + dis	**93.9**	**76.3**	**81.2**	46.5	43.3	**50.1**
DialBERT + cov + layer	93.2	72.0	79.5	43.1	40.0	46.8
Ptr-Net	92.3	70.2	-	36.0	33.0	38.9
Ptr-Net + Joint train&Self-link	94.2	80.1	-	44.5	44.9	44.2
Structural Characterization	94.6	76.8	84.2	51.7	51.8	51.7

Table 3: Results on the Ubuntu IRC development and test sets. Note that feature was introduced along with the original dataset (Kummerfeld et al., 2019), so the "feature" used with different models was the same. The results marked with * were copied from their corresponding publications. Dec. Att. denoted the decomposable attention model (Parikh et al., 2016), ESIM denoted the enhanced sequential inference model (Chen et al., 2017), and MHT denoted masked hierarchical Transformer (Zhu et al., 2020). Numbers in bold denoted the best performance without comparing with Ptr-Net (Yu and Joty, 2020) and structural characterization(Ma et al., 2022), which are the latest proposed methods for dialogue disentanglement and are included for reference.

DialBERT + feature: The same setting as DialBERT, but also combined with the features used in Kummerfeld et al. (2019). The features consist of three parts: (1) Global-level features, including year and frequency of the conversation. (2) Utterance level features, including types of message, targeted or not, time difference between the last message, etc. (3) Utterance pair features including how far apart in position and the time between the messages, whether one message targets another, etc. Specifically, we concatenate these external features with high-order feature vectors **F** in our model. These features are same as those used in other baseline models.

Model	Graph	Conversation
ESIM	23.2	0
Decomposable Attention	16.4	0
BERT	29.6	0.24
DialBERT	33.7	0.36
DialBERT + cov	34.5	0.31
DialBERT + cov + uni	**36.1**	0.38
DialBERT + cov + dis	34.4	**0.41**
DialBERT + cov + layer	33.1	0.29

Table 4: Results of different models on the Reddit test set in terms of the accuracy (%).

	VI	ARI	1-1	F1	P	R
Our model	**93.9**	**76.3**	**81.2**	**46.5**	**43.3**	**50.1**
- extra losses	92.7	69.2	78.5	44.3	42.1	46.7
- adaptation	92.5	67.8	78.6	41.0	37.6	45.1
- BiLSTM	90.8	62.9	75.0	32.5	29.3	36.6

Table 5: Ablation analysis on different components using the Ubuntu IRC dataset.

DialBERT + ensemble: In this setting, the weights of the model prediction probability were averaged for each sample across 8 DialBERT models.

DialBERT w/o. adaptation: In this setting, the adaptation process was ablated. DialBERT was finetuned on the IRC dataset directly.

DialBERT + cov: The conversation-structure loss was employed in addition to the cross-entropy loss.

DialBERT + cov + (uni or dis or layer): Three results of using different tree-structure losses were reported.

5.5 Experimental Results

The performances of different models on the IRC test set are shown in Table 3. Our model outperforms all of the previous models in all evaluation metrics. Specifically, on the test set, the previous work using an ensemble of 10 feedforward models obtained through a vote is capable of reaching the previous best performance. We can see that our best model (DialBERT+cov+dis) achieves better performance by a large margin. To compare our results with those reported in Zhu et al. (2020), we report the performances of DialBERT and DialBERT *w/o. adaptation* on the development set as well. [4] We can see even without domain

[4] Zhu et al. (2020) did not include results on the test set.

Parent	DialBERT	DialBERT extra losses	Index	Message
...	...	...		
1232	1232	1232	1232	[19:15] franendar: how can I install a specific glibc version?
1226	1226	1226	1233	[19:15] EriC: paste grep Prompt /etc/update-manager/release-upgrades
1232	1232	1232	1234	[19:15] franendar: im getting this: sudo apt-get install build-essential
1233	1233	1233	1235	[19:15] EriC: empty
1232	1236	**1232**	1236	[19:15] MonkeyDust: many glibc questions these days, i wonder how come
1234	1234	1234	1237	[19:15] franendar: im getting this: version 'GLIBCXX_3.4.21' not found
1235	1235	1235	1238	[19:15] EriC: cat /etc/update-manager/release-upgrades
1223	1231	1231	1239	[19:15] nick420: Unable to locate package java8-installer
1238	1238	1238	1240	[19:16] EriC: prompt=never
1240	1240	1240	1241	[19:16] EriC: So, prompt=lts?
1241	1240	**1241**	1242	[19:16] EriC: yeah
1242	1242	1242	1243	[19:16] EriC: Thanks
...	...	...		

Table 6: An example that DialBERT cannot predict correctly, but DialBERT + *extra losses* does. In this table, Parent is the golden label; DialBERT and DialBERT + *extra losses* is the the perdiction of different models; Index is the message index.

adaptation and extra losses, DialBERT already outperforms *MHT+feature*. All our other models perform even better on the development set, but due to the space limit, we only report the above two models on the development set.

The same observation can be seen on the Reddit dataset as shown in Table 4. Note that the values of conversation accuracy (*Conv. Acc.*) are small, due to the definition of the metric itself.

Different from other NLP tasks, according to the results, BERT does not have much advantages over other models, which indicates semantic knowledge learned from pre-training is not a direct indicator of improvement for disentanglement. The result that DialBERT outperforms BERT on all six evaluation metrics could be explained by the vital importance of context in conversations disentanglement, and DialBERT makes better use of pre-trained knowledge. The substantial margin between DialBERT and DialBERT *w/o. adaptation* demonstrates adaptation does give further improvement of DialBERT. It is also notable that DialBERT+*feature* does not have much performance improvement compared with DialBERT, which means the information contained in feature has been implicitly learned during the domain adaptation process. As the result, we further report the ensemble results and external loss results based on DialBERT with adaptation.

The results that DialBERT+*cov* outperforms DialBERT shows that the conversation-structure loss does help. Among the three strategies of tree-structure losses, only the *penalty-by-distance* strategy can further improve the performance of DialBERT+*cov*. The reason might be both *uniform-penalty* strategy and *penalty-by-layer-difference* strategy ignore the distance between each message and target message in tree structures, and distance information is of vital importance to understand the conversation structures. That explains why *penalty-by-distance* strategy can further improve the result in both the IRC test set and in Reddit test set.

It can be seen that the results of DialBERT and DialBERT with conversation-structure loss doesn't show a substantial margin in the Reddit test set. The reason might be the differences in data collection. For the IRC dataset, data are collected from *Linux IRC channel* which means different conversations can happen at the same time and messages in context range are not necessary within the same conversation with the target message. But for the Reddit dataset, data are crawled by a list of all posts in a conversation which means messages of each conversation are together in the dataset. As a result, the conversation information can not give as much improvement as in IRC dataset.

5.6 The Value Design for Tree-Structure Loss

The selection of d and l is based on the statistic of both datasets that we used. For equation 9, $d = 5$ will cover most of samples. Because our target is to find the parent message of target message. So we set $d = 0$ a smaller value to give the "real" parent message more "credit". For the same reason, we set threshold d and l to be 20 and 10 in equation 11 and equation 12 respectively. Please note that the d in equation 9 is designed for ancestor messages. The d in equation 11, however, is designed for non-ancestor messages which are generally further away from the target message. The d in equation 11 will not be 0. As the result, we set different threshold d value. The intention that we designed descending y_i^b based on distance (or layer-difference) is the assumption that the nearer a message and the target message is the more semantic relevant it could be. We designed the *uniform-penalty* strategy to verify the correctness of the assumption (as shown in Table 3), and results show that *penalty-by-distance* and *penalty-by-layer-difference* do reach better performance.

5.7 Ablation

To find out how each component contributes to the final results, we display the ablation analysis of different component based on our best system DIALBERT+*cov*+*dis* (as shown in Table 5). The performance of the model drops in all of 6 evaluation metrics after the removal of extra losses, which demonstrates the effectiveness of integrating conversation structure information into the losses.

Moreover, the performance of the model drops in 5 out of 6 evaluation metrics after the removal of adaptation process, which indicates adaptation learns useful semantic information, especially under the condition that the dataset is in a specific domain. After the removal of BiLSTM, in which the model has to make a prediction without any context consideration, results fall remarkably according to all evaluation metrics. As we discussed before, context is very important for disentangling a conversation. We can see from the ablation results, every component added on BERT in our model contributes to the final result.

Our model can not only introduce global and local conversation semantics but also introduce the conversation structures implicitly, resulting in achieving a new state-of-the-art results by outperforming other models substantially.

5.8 Case Study

As shown in Table 6, there are three conversations involve in this example, i.e., {1232, 1234, 1236, 1237 }, {1233, 1235, 1238, 1240, 1241, 1242, 1243} and {1249}, where these numbers denote the index for each message. For the messages 1236 and 1242, DIALBERT + *extra losses* can find the correct parent message, which indicates that extra losses do help the DIALBERT in dialogue disentanglement. Specifically, for the message 1236, conversation-structure loss plays a more important role, because the preceding messages after parent message are from two conversation. For the message 1242, tree-structure loss plays a more important role, because the preceding messages after parent message are from the same conversation. For message 1239, both DIALBERT and DIALBERT + *extra losses* cannot predict correctly, the reason might be that the distance from parent message is too far in this case, which demonstrates that dialogue disentanglement is still hard and extra losses can not handle all the cases.

6 Conclusions

In this paper, we propose a novel framework for dialogue disentanglement. Different from previous work, we integrate both local and global semantics by proposing an adapted hierarchical BERT-based model (DIALBERT) to disentangle conversations. Moreover, in order to make use of conversation structures, we finetune our model with two losses (i.e., conversation-structure loss and tree-structure loss). We evaluate our method on two large datasets. Results show that our method achieves a new state-of-the-art performances on both datasets and outperforms models from previous work with a substantial margin. In the future, we will design non-heuristic methods for modeling the conversation structure with less hyperparameters which is a challenge worth exploring.

Acknowledgements

We thank anonymous reviewers for their valuable comments.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Paige H Adams and Craig H Martell. 2008. Topic detection and extraction in chat. In *2008 IEEE international conference on Semantic computing*, pages 581–588. IEEE.

Paul M Aoki, Matthew Romaine, Margaret H Szymanski, James D Thornton, Daniel Wilson, and Allison Woodruff. 2003. The mad hatter's cocktail party: a social mobile audio space supporting multiple simultaneous conversations. In *Proceedings* *of the SIGCHI conference on human factors in computing systems*, pages 425–432. ACM.

Qian Chen, Zhen-Hua Ling, and Xiaodan Zhu. 2018. Enhancing sentence embedding with generalized pooling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1815–1826.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

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.

Wenchao Du, Pascal Poupart, and Wei Xu. 2017. Discovering conversational dependencies between messages in dialogs. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Micha Elsner and Eugene Charniak. 2008. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842.

Micha Elsner and Eugene Charniak. 2010. Disentangling chat. *Computational Linguistics*, 36(3):389–409.

Micha Elsner and Eugene Charniak. 2011. Disentangling chat with local coherence models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1179–1189.

Micha Elsner and Warren Schudy. 2009. Bounding and comparing methods for correlation clustering beyond ilp. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 19–27.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020a. Speaker-aware BERT for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2041–2044.

Jia-Chen Gu, Tianda Li, Quan Liu, Xiaodan Zhu, Zhen-Hua Ling, and Yu-Ping Ruan. 2020b. Pre-trained and attention-based neural networks for building noetic task-oriented dialogue systems. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, Workshop on the Eighth Dialog System Technology Challenge, DSTC8, New York, NY, USA, February 7-12, 2020.*

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.

Jyun-Yu Jiang, Francine Chen, Yan-Ying Chen, and Wei Wang. 2018. Learning to disentangle interleaved conversational threads with a siamese hierarchical network and similarity ranking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1812–1822.

Jonathan K Kummerfeld, Sai R Gouravajhala, Joseph J Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856.

Hui Liu, Zhan Shi, Jia-Chen Gu, Quan Liu, Si Wei, and Xiaodan Zhu. 2021a. End-to-end transition-based online dialogue disentanglement. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3868–3874.

Hui Liu, Zhan Shi, and Xiaodan Zhu. 2021b. Unsupervised conversation disentanglement through co-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2345–2356.

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*.

Elijah Mayfield, David Adamson, and Carolyn Rose. 2012. Hierarchical conversation structure prediction in multi-party chat. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 60–69.

Shikib Mehri and Giuseppe Carenini. 2017. Chat disentanglement: Identifying semantic reply relationships with random forests and recurrent neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 615–623.

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

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

David Traum. 2004. Issues in multiparty dialogues. In *Workshop on Agent Communication Languages*, pages 201–211. Springer.

David C Uthus and David W Aha. 2013. Multiparticipant chat analysis: A survey. *Artificial Intelligence*, 199:106–121.

Lidan Wang and Douglas W Oard. 2009. Context-based message expansion for disentanglement of interleaved text conversations. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 200–208.

Yi-Chia Wang, Mahesh Joshi, William W Cohen, and Carolyn Penstein Rosé. 2008. Recovering implicit thread structure in newsgroup style conversations. In *Proceedings of the Second International Conference on Weblogs and Social Media, ICWSM 2008*.

Taesun Whang, Dongyub Lee, Chanhee Lee, Kisu Yang, Dongsuk Oh, and Heuiseok Lim. 2020. An effective domain adaptive post-training method for BERT in response selection. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2020, pages 1585–1589.

Deunsol Yoon, Dongbok Lee, and SangKeun Lee. 2018. Dynamic self-attention: Computing attention over words dynamically for sentence embedding. *arXiv preprint arXiv:1808.07383*.

Tao Yu and Shafiq Joty. 2020. Online conversation disentanglement with pointer networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6321–6330.

Henghui Zhu, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Who did they respond to? conversation structure modeling using masked hierarchical transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9741–9748.