MUIDIAL: Improving Dialogue Disentanglement with Intent-Based Mutual Learning

Ziyou Jiang[1,4], **Lin Shi**[1,4*], **Celia Chen**[5], **Fangwen Mu**[1,4], **Yumin Zhang**[1,4] and **Qing Wang**[1,2,3,4*]

[1]Laboratory for Internet Software Technologies, Institute of Software Chinese Academy of Sciences
[2]State Key Laboratory of Computer Sciences, Institute of Software Chinese Academy of Sciences
[3]Science&Technology on Integrated Information System Laboratory, Institute of Software Chinese Academy of Sciences
[4]University of Chinese Academy of Sciences, Beijing, China
[5]Department of Computer Science, Occidental College, Los Angeles, CA, USA
{ziyou2019, shilin, fangwen2020,yumin2020, wq}@iscas.ac.cn, qchen2@oxy.edu

Abstract

The main goal of dialogue disentanglement is to separate the mixed utterances from a chat slice into independent dialogues. Existing models often utilize either an utterance-to-utterance (U2U) prediction to determine whether two utterances that have the "reply-to" relationship belong to one dialogue, or an utterance-to-thread (U2T) prediction to determine which dialogue-thread a given utterance should belong to. Inspired by mutual leaning, we propose MUIDIAL, a novel dialogue disentanglement model, to exploit the intent of each utterance and feed the intent to a mutual learning U2U-U2T disentanglement model. Experimental results and in-depth analysis on several benchmark datasets demonstrate the effectiveness and generalizability of our approach.

1 Introduction

Online communication platforms such as Gitter and Slack have been widely adopted by developers as the main communication channel to discuss events, issues, tasks, and personal experiences. Despite the convenience of sharing and collaborating in real-time, there are often multiple conversations occurring among a large group of participants concurrently, thus making it difficult to detect topics, analyze user behaviors and summarize contents automatically.

Various automatic dialogue disentanglement approaches have been proposed with the goal to separate utterances into dialogues and better facilitate the usage of massive live chat data. These approaches share two common aspects to achieve dialogue disentanglement [Zhu *et al.*, 2021]. One aspect is the utterance-to-utterance prediction (U2U), which predicts the "reply-to" probability between two utterances. U2U can find utterances with similar semantic features, which improves the accuracy of disentanglement. The other aspect is the utterance-to-thread prediction (U2T) that predicts which dialogue-thread a given utterance belongs to. U2T can analyze some weak semantic related utterances (e.g., "Greeting" and "Feedback"), thus improving the completeness of dialogues.

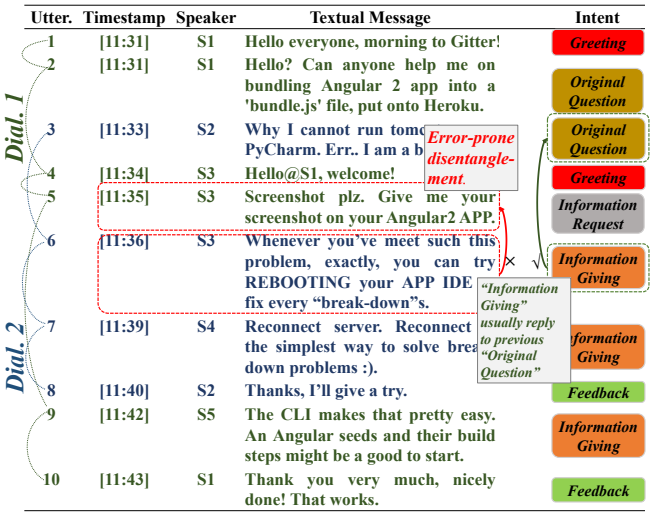

Figure 1: An example of dialogue disentanglement in the Gitter dataset, including timestamp, speaker, and textual message. The curves with different colors represent the links of different dialogues after disentanglement.

Although there are some hybrid approaches that aim to combine U2U and U2T predictions together [Tan *et al.*, 2019; Liu *et al.*, 2021; Pappadopulo *et al.*, 2021], the two predictions are still trained independently, and the learning is hardly shared between them comprehensively.

In this paper, we propose MUIDIAL, a novel intent-based dialogue disentanglement with mutual learning. **First**, we enhance the utterance embedding by integrating the intent embedding and five heuristic features with textual utterance embedding. Figure 1 shows how the user-intent can benefit the disentanglement performance. As shown in the figure, it is difficult to distinguish between u_5 and u_6 in terms

*Corresponding Author

of the traditional semantic features (e.g., timestamp, speaker or text message). However, if we consider intent, "Information Giving" is usually used to reply to "Original Question", not "Information Giving", which means u_6 and u_3 tend to be disentangled into the same dialogue, rather than u_5. **Second**, we design two new types of utterance prediction components, U2UDIAL and U2TDIAL, where U2UDIAL leverages the pointer mechanism to predict the "reply-to" relationships and U2TDIAL leverages the attention-based state transition mechanism to predict the "belong-to" relationships. **Finally**, we utilize a new mechanism to share the learning ability between U2UDIAL and U2TDIAL via mutual learning [Zhang *et al.*, 2018], which is found successful in several AI research areas [Liu *et al.*, 2019; Yang *et al.*, 2020] but has not been commonly facilitated in dialogue disentanglement.

To evaluate the effectiveness of our approach, we conduct an exploratory study on four cross-domain benchmark datasets: IM (Instant Messenger), Reddit, IRC (Ubuntu Internet Relay Chat), and Gitter. The results show that MUIDIAL outperforms SOTA baselines on all the datasets.

Our major contributions are summarized as follows:

- We enhance the utterance embedding for dialogue disentanglement with user intents and five heuristic features.
- We adopt mutual learning in the dialogue disentanglement setting, where we propose MUIDIAL, a novel intent-based dialogue disentanglement model, which takes the advantages of both U2U and U2T predictions with mutual distillation.
- We demonstrate the effectiveness and efficiency of MUIDIAL on commonly used benchmark datasets.

2 Methodology

There are four main steps to construct MUIDIAL, as shown in Figure 2, including: 1) the intent-based representative learning to embed the input utterances; 2) multiple thread probability computation using both U2U and U2T predictions; 3) mutual learning to optimize parameters for both predictions; and 4) mutual predicting to output convergent disentanglement results via voting.

2.1 Intent-based Utterance Embedding

We first embed utterances with text, heuristic, and user-intent encoders; then we learn its context into the context-aware utterance embedding to obtain a rich representation.

Individual Utterance Embedding

Given a set of utterances u_i, a slice of multi-party chat is defined as $[u_1, u_2, ..., u_n]$. Each utterance is a tuple $u_i = \langle t_i, s_i, \boldsymbol{w}_i \rangle$, where $\boldsymbol{w}_i = [w_{i1}, w_{i2}..., w_{im}]$ is the word sequence posted by speaker s at the time t. The dialogues in one chat slice are noted as $\boldsymbol{D} = [\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_K]$, and each utterance is associated with a dialogue-thread $\boldsymbol{D}_k$. We learn the representative of individual utterance u_i with three encoders ($\boldsymbol{u}_i$, $\boldsymbol{r}_i$, and $\boldsymbol{\epsilon}_i$), and concatenate them as the individual utterance embedding $\boldsymbol{\xi}_i = [\boldsymbol{u}_i; \boldsymbol{r}_i; \boldsymbol{\epsilon}_i]$.

Code	Label	Description of labels
OQ	Original Question	Speaker proposes the first question to initialize a dialogue.
FQ	Follow-Up Question	Speaker raises follow-up questions about the related issues.
IG	Information Giving	Speaker provides some information to other speakers.
IS	Information Seeking	Speaker seeks more information from other speakers.
FB	Feedback	Speaker provides reactions/information to solutions posted by other speakers.
OT	Others	Greetings, junk messages or other uncategorized utterances.

Table 1: The categories of user intents.

Text Encoder. Given the word sequence $\boldsymbol{w}_i$ in utterance u_i, we embed it into $\boldsymbol{u}_i$ by using pre-trained BERT [Devlin *et al.*, 2019] model.

$$\boldsymbol{u}_i = \mathrm{BERT}\left([w_{i1}, w_{i2}, ..., w_{im}]\right) \tag{1}$$

Heuristic Feature Encoder. To enrich the utterance embedding with high-level semantic information, we extract five predefined features from each utterance $u_i = \langle t_i, s_i, \boldsymbol{w}_i \rangle$ in a heuristic manner.

1. Speaker: $\boldsymbol{s}_i = [\mathbb{I}\left(k = \mathrm{Index}\left(s_i\right)\right)]$, where $\mathbb{I}$ is the indicator function that returns 1 if the element subscript of $\boldsymbol{s}_i$ is consistent with the identifier of s_i.
2. Time Difference: $\Delta t_i = t_i - t_0$, where t_i is the timestamp of the current utterance, and t_0 is the timestamp of the first utterance in the same chat.
3. Topic: $\boldsymbol{\tau}_i = \mathrm{GloVe}\left(\mathrm{LDA}\left(\boldsymbol{w}_i\right)\right)$, where LDA is the topic extraction model pre-trained on 10K topics, and GloVe is the word embedding model pre-trained on 30K words.
4. Entity: $\boldsymbol{e}_i = \mathrm{GloVe}\left(\text{LSTM-CRF}\left(\boldsymbol{w}_i\right)\right)$, where LSTM-CRF is the bi-directional entity extraction model, and GloVe is the pre-trained word embedding model.
5. Mention: $\boldsymbol{m}_i = \mathbf{M}\left(s_i\right)$, where mention history $\mathbf{M}$ is a diagonal matrix for all speakers [Yu and Joty, 2020]. We set the $\mathbf{M}(s_i, s_j)$ as the frequency that speakers s_i and s_j mention each other.

The output of the heuristic feature encoder $\boldsymbol{r}_i$ is the concatenation of the five features:

$$\boldsymbol{r}_i = [\boldsymbol{s}_i; \Delta t_i; \boldsymbol{\tau}_i; \boldsymbol{e}_i; \boldsymbol{m}_i] \tag{2}$$

User-intent Encoder. We use the concatenation of text embedding $\boldsymbol{u}_i$ and heuristic features $\boldsymbol{r}_i$ as the input of user-intent encoder. First, we use a BiLSTM [Hochreiter and Schmidhuber, 1997] model to encode the utterance information into a deep contextual representation $\boldsymbol{h}_i^{\epsilon}$. Then, we use a multi-layer perceptron (MLP), which includes three fully-connected layers (FCNN) and uses ReLU as the activation function to encode the user-intent embedding $\boldsymbol{\epsilon}_i$.

$$\overleftarrow{\boldsymbol{h}}_i^{\epsilon}, \overrightarrow{\boldsymbol{h}}_i^{\epsilon} = \mathrm{BiLSTM}\left([\boldsymbol{u}_i; \boldsymbol{r}_i]\right), \boldsymbol{\epsilon}_i = \mathrm{MLP}\left([\overleftarrow{\boldsymbol{h}}_i^{\epsilon}; \overrightarrow{\boldsymbol{h}}_i^{\epsilon}]\right) \tag{3}$$

Following the previous user-intent study [Qu *et al.*, 2018], we define six intents that commonly appear in live chats, as described in Table 1. The output of user-intent encoder $\boldsymbol{\epsilon}_i$ is a 6-dimensional vector that records the weights of the six

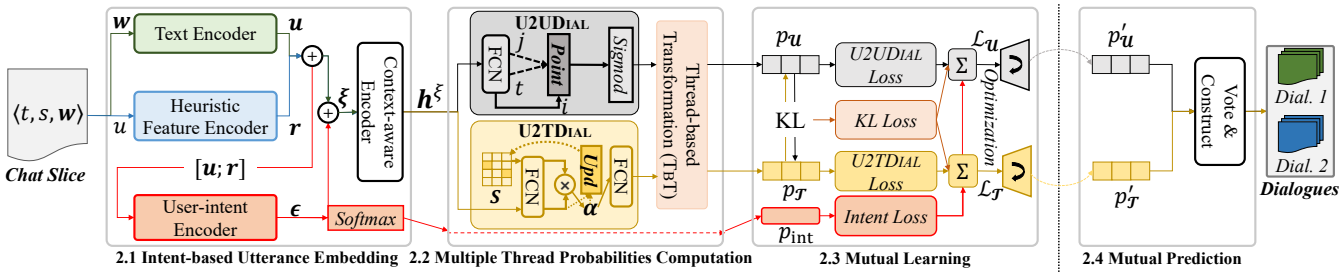

Figure 2: The architecture of MUIDIAL.

user intents according to each dimension. Finally, we use the Softmax function to predict the intent probability:

$$p_{\text{int}}(u_i) = p([\text{OQ}, \text{FQ}, \text{IG}, \text{IS}, \text{FB}, \text{OT}]\,|u_i) = \text{Softmax}\,(\boldsymbol{\epsilon}_i) \quad (4)$$

where $p_{\text{int}}(u_i)$ indicates the probability of intent prediction.

Context-aware Encoder

This component aims to account for representing the full utterance context in one chat slice. Given a sequence of individual utterance embeddings $[\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, ..., \boldsymbol{\xi}_n]$ in one chat slice, we use a BiLSTM model to further embed the contextual information for each utterance u_i.

$$\overrightarrow{\boldsymbol{h}}_i^{\xi}, \overleftarrow{\boldsymbol{h}}_i^{\xi} = \text{BiLSTM}\,(\boldsymbol{\xi}_i)\,, \boldsymbol{h}_i^{\xi} = [\overrightarrow{\boldsymbol{h}}_i^{\xi}, \overleftarrow{\boldsymbol{h}}_i^{\xi}] \quad (5)$$

where $\boldsymbol{h}_i^{\xi}$ is the output of the intent-based utterance embedding for utterance u_i.

2.2 Multiple Thread Probabilities Computation

Given the intent-based embedding of all the utterances in the chat slice $[\boldsymbol{h}_1^{\xi}, ..., \boldsymbol{h}_n^{\xi}]$ as input, we first construct U2UDIAL and U2TDIAL to respectively output the "reply-to" and "belong-to" probability matrices. Since the two probability matrices have different dimensions, we then design the TBT algorithm to transform them into the same dimension.

Utterance-to-Utterance Prediction (U2UDIAL)

U2UDIAL uses the pointer module [Nguyen *et al.*, 2020] to predict the "reply-to" probability between the current u_i and its former utterances $u_{j\leq i}$. Given a sequence of the intent-based utterance embedding $[\boldsymbol{h}_1^{\xi}, \boldsymbol{h}_2^{\xi}, ..., \boldsymbol{h}_n^{\xi}]$, where $\boldsymbol{h}_i^{\xi} \in \mathbb{R}^{1\times\delta}$ (δ indicates the size of the dialogue embedding vector), we use a fully-connected neural network (FCNN) to map the hidden representation to new hidden vectors. We then use the dot-product to calculate the similarity and normalize the result with Sigmod activation:

$$\mathbf{U}\,(u_i, u_j) = \begin{cases} 1.0, & \text{if } 1 \leq j = i \leq n \\ \text{Sigmoid}(\boldsymbol{h}_i^{\xi}\mathbf{W}_{\mathcal{U}}\boldsymbol{h}_j^{\xi\top}), & \text{if } 1 \leq j < i \leq n \end{cases} \quad (6)$$

where $\mathbf{W}_{\mathcal{U}} \in \mathbb{R}^{\delta\times\delta}$ represents the trainable parameters of FCNN and the "reply-to" probability matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$ indicates the probability that $u_i \rightarrow u_j$ is a "reply-to" relationship.

Utterance-to-Thread Prediction (U2TDIAL)

U2TDIAL uses the attention-based state-transition mechanism [Chen and Manning, 2014] to predict the "belong-to" probability. Given the intent-based utterance embedding $\boldsymbol{h}_i^{\xi}$, we randomly initialize $\boldsymbol{S}_1 \in \mathbb{R}^{K\times\delta}$ as $\boldsymbol{S}_1 = [\boldsymbol{S}_{11}, \boldsymbol{S}_{12}, ..., \boldsymbol{S}_{1K}]$, where K is the number of dialogue-threads in a chat slice and $\boldsymbol{S}_{1k}$ is the sub-vector to represent the feature of one dialogue cluster. At timestamp i, we choose the dot-product and Softmax to measure the similarity between $\boldsymbol{h}_i^{\xi}$ and its state $\boldsymbol{S}_i$, and input the result to another FCNN to predict the "belong-to" probability.

$$\mathbf{T}\,(u_i) = \mathbf{W}_{\mathcal{T}}'\boldsymbol{\alpha}_i = \mathbf{W}_{\mathcal{T}}'\left[\text{Softmax}(\boldsymbol{S}_i\mathbf{W}_{\mathcal{T}}\boldsymbol{h}_i^{\xi\top})\right] \quad (7)$$

where matrix $\mathbf{W}_{\mathcal{T}} \in \mathbb{R}^{\delta\times\delta}$ represents a set of trainable parameters of coupling measurement, and $\mathbf{W}_{\mathcal{T}}' \in \mathbb{R}^{K\times K}$ is the trainable parameter of FCNN. The k_{th} element of the weight vector $\boldsymbol{\alpha}_i \in \mathbb{R}^{K\times 1}$ denotes the attention weight between u_i and dialogue-thread $\boldsymbol{D}_k$. $\mathbf{T} \in \mathbb{R}^{n\times K}$ indicates the probability that u_i belongs to the thread $\boldsymbol{D}_k$. Finally, we use $\boldsymbol{\alpha}_i$ to weight $\boldsymbol{h}_i^{\xi}$ and update $\boldsymbol{S}_i$ with self-attention mechanism [Yang *et al.*, 2016] to aggregate the previous embedding:

$$\begin{aligned} \gamma_j &= \frac{\exp\left[\tanh\left(\mathbf{W}_{\text{attn}}(\boldsymbol{\alpha}_j\boldsymbol{h}_j^{\xi})^{\top} + \mathbf{b}_{\text{attn}}\right)\right]}{\sum_{1\leq j\leq i}\exp\left[\tanh\left(\mathbf{W}_{\text{attn}}(\boldsymbol{\alpha}_j\boldsymbol{h}_j^{\xi})^{\top} + \mathbf{b}_{\text{attn}}\right)\right]} \\ \boldsymbol{S}_{i+1} &= \sum_{1\leq j\leq i}\gamma_j(\boldsymbol{\alpha}_j\boldsymbol{h}_j^{\xi}) \end{aligned} \quad (8)$$

where $\mathbf{W}_{\text{attn}} \in \mathbb{R}^{k\times\delta}$ and $\mathbf{b}_{\text{attn}} \in \mathbb{R}^{K\times 1}$ are trainable parameters for calculating the attention weights.

Thread-based Transformation (TBT)

Given the two probability matrices $\mathbf{U}$ and $\mathbf{T}$ outputted by U2UDIAL and U2TDIAL, the TBT function can be defined as $\text{TBT}\,(\{\mathbf{U}, \mathbf{T}\}) \rightarrow p(u_i, u_j)$, where $p(u_i, u_j)$ indicates the probability of whether the current utterance u_i and the former utterances u_j should be classified into the same dialogue-thread. Figure 3 visualizes the details of TBT function, which consists $\text{TBT}_{\mathcal{U}}$ and $\text{TBT}_{\mathcal{T}}$. For "reply-to" probability $\mathbf{U}$, $\text{TBT}_{\mathcal{U}}$ transforms the U2UDIAL thread probability with an iterative dynamic programming algorithm:

$$p_{\mathcal{U}}\,(u_i, u_j) = \begin{cases} 1.0, & \text{if } 1 \leq j = i \leq n \\ \dfrac{1}{i-j}\displaystyle\sum_{j\leq t<i}\mathbf{U}\,(u_i, u_t)\,p_{\mathcal{U}}\,(u_t, u_j)\,, & \text{if } 1 \leq j < i \leq n \end{cases} \quad (9)$$

For "belong-to" probability $\mathbf{T}$, $\text{TBT}_{\mathcal{T}}$ transforms the U2TDIAL thread probability with dot-product:

$$p_{\mathcal{T}}\,(u_i, u_j) = \begin{cases} 1.0, & \text{if } 1 \leq j = i \leq n \\ \mathbf{T}(u_i)\mathbf{T}(u_j)^{\top} = \left(\mathbf{T}\mathbf{T}^{\top}\right)_{ij}, & \text{if } 1 \leq j < i \leq n \end{cases} \quad (10)$$

The output probabilities of TBT include both $\mathbb{R}^{n\times n}$ lower triangular matrices, with 1 on the diagonal. The other element values are distributed in $(0, 1)$.

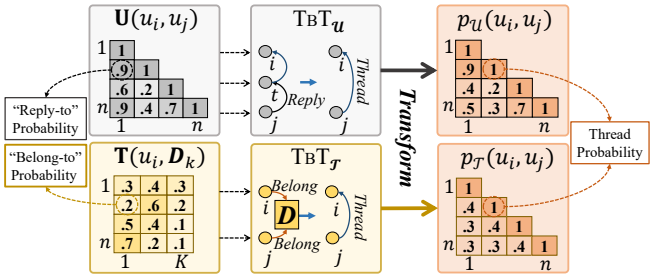

Figure 3: Description of thread-based transformation (TBT). TBT$_{\mathcal{U}}$: If (u_i, u_t) belongs to the same thread and $u_j \rightarrow u_t$ is the "reply-to" relationship, then (u_i, u_j) belongs to the same thread. TBT$_{\mathcal{T}}$: If u_i and u_j both belong to $\boldsymbol{D}_k$, then (u_i, u_j) belongs to the same thread.

2.3 Mutual Learning

We calculate four losses with two probabilities and optimize the MUIDIAL mutually. The goal is to optimize the intent embedding together with the thread probability prediction.

Loss Functions

The four losses include intent loss, U2UDIAL loss, U2TDIAL loss, and KL loss.

Intent Loss. Given the intent probabilities produced by the user-intent encoder $p_{\text{int}}(u_i)$, we use the cross-entropy loss to calculate the intent loss $\mathcal{L}_{\text{int}}$:

$$\mathcal{L}_{\text{int}} = -\frac{1}{T \times n} \sum_{1 \le t \le T} \sum_{1 \le i \le n} H\left(Y_i^{\text{int}}, p_{\text{int}}(u_i)\right) \tag{11}$$

where $H(\cdot,\cdot)$ is the cross-entropy function, Y_i^{int} is the truth intent label of u_i, and T is the batchsize of training datasets.

U2UDIAL and U2TDIAL Losses. Similarly, we use the Cross-Entropy loss to calculate the U2UDIAL loss $\mathcal{L}_{\mathcal{U}}^{\text{thd}}$ and U2TDIAL loss $\mathcal{L}_{\mathcal{T}}^{\text{thd}}$ with thread probabilities.

$$\begin{aligned}
\mathcal{L}_{\mathcal{U}}^{\text{thd}} &= -\frac{2}{T \times n(n+1)} \sum_{1 \le t \le T} \sum_{1 \le j \le i \le n} H\left(Y_{ij}^{\text{thd}}, p_{\mathcal{U}}(u_i, u_j)\right) \\
\mathcal{L}_{\mathcal{T}}^{\text{thd}} &= -\frac{2}{T \times n(n+1)} \sum_{1 \le t \le T} \sum_{1 \le j \le i \le n} H\left(Y_{ij}^{\text{thd}}, p_{\mathcal{T}}(u_i, u_j)\right)
\end{aligned} \tag{12}$$

where Y_{ij}^{thd} indicates the truth label of whether u_i and its former utterance $u_{j \le i}$ are in the same dialogue-thread.

KL Loss. In order to measure the mutual difference between two output distributions, we calculate Kullback-Leibler (KL) divergence between two thread-based probabilities as the KL loss:

$$\mathcal{L}_{\text{KL}} = \frac{2}{T \times n(n+1)} \sum_{1 \le t \le T} \sum_{1 \le j \le i \le n} KL\left(p_{\mathcal{U}}(u_i, u_j) \,||\, p_{\mathcal{T}}(u_i, u_j)\right) \tag{13}$$

where KL $(\cdot||\cdot)$ is the function of KL divergence.

Mutual Optimization

With the above losses, we iteratively optimize the parameters of MUIDIAL by performing the two optimizations mutually:

- Optimize $\mathcal{L}_{\mathcal{U}} = \alpha\mathcal{L}_{\text{int}} + \beta\mathcal{L}_{\text{KL}} + (1-\alpha-\beta)\mathcal{L}_{\mathcal{U}}^{\text{thd}}$ where α and β are the loss-balancing parameters.
- Optimize $\mathcal{L}_{\mathcal{T}} = \alpha\mathcal{L}_{\text{int}} + \beta\mathcal{L}_{\text{KL}} + (1-\alpha-\beta)\mathcal{L}_{\mathcal{T}}^{\text{thd}}$, where α and β are the loss-balancing parameters.

We employ gradient descent to minimize $\mathcal{L}_{\mathcal{U}}$ and $\mathcal{L}_{\mathcal{T}}$. The two optimization processes are repeated simultaneously until the model is convergent.

2.4 Mutual Prediction

Since $p_{\mathcal{U}}(u_i, u_{j \le i})$ and $p_{\mathcal{T}}(u_i, u_{j \le i})$ may give different disentanglement predictions, we adopt a vote&construct strategy to determine the final disentangled dialogues. Given the multi-party chat slice $[u'_1, u'_2, ..., u'_n]$ in the test dataset, we separately predict the probabilities $p_{\mathcal{U}}(u'_i, u'_{j \le i})$ and $p_{\mathcal{T}}(u'_i, u'_{j \le i})$. Then, we predict that u'_i and $u'_{j \le i}$ are in the same thread if and only if both probabilities satisfy $p(u'_i, u'_{j \le i}) > \eta$, where η is the cutoff-value to separate the positive and negative classes. We follow the existing works to tune η until MUIDIAL achieves the optimal performance, and choose the cutoff-value as $\eta = 0.6$. Finally, we use the predicted relationships to construct independent dialogues $\boldsymbol{D}$ and output the disentangled results.

3 Experiments

3.1 Experiment Settings

Datasets. We adopt four benchmark datasets for the dialogue disentanglement task. As shown in Table 3, IM and Reddit contain social chats collected from Microsoft Messenger and Reddit forum respectively, whereas IRC [Kummerfeld *et al.*, 2019] and Gitter focus on technical chats collected from Ubuntu IRC chat log and Gitter chat rooms from eight open-source communities respectively.

Ground-truth Labeling for Intent. We manually label the intent category for each utterance on the sampled training and validate dataset from the four benchmark datasets, as shown in the last row in Table 3. To guarantee the correctness of the labeling results, we carefully assemble the labeling team with four members (2PhD/1MS students and 1 research fellow) and develop a widely accepted process in our annotation. Cohen's Kappa is measured to indicate the level of agreement among team members. The average Cohen's Kappa score is 0.84, which proves the annotation agreement[1].

Evaluation Metrics. We use four commonly used metrics to evaluate the dialogue disentanglement results: Normalized Mutual Information (NMI) [Strehl and Ghosh, 2002], Adjusted Rand Index (ARI) [Santos and Embrechts, 2009], F1-score (F1), and Dialogue Levenshtein-Distance (DLD) [Jiang *et al.*, 2021]. ARI is the most strict metric that bases the evaluation on pairwise biases, while NMI penalizes more on the cluster-level. F1 is the reconciliation of accuracy and recall and DLD calculates Levenshtein distance of correcting negative disentanglement, which reflects the user satisfaction.

Implementation Details. The entire chat log is first processed. We split a chat log into a set of chat slices if 1) the number of dialogues in the chat exceeds 4, or 2) the time interval between two adjacent utterances exceeds 48 hours. If the number of utterances in a set of chat slices exceeds 100, a new set of chat slices is created. DLD is reported to be a

[1] According to magnitude guidelines, kappa value above 0.81 is considered as almost perfect agreement.

Models \ Metrics				IM				Reddit				IRC				Gitter				*Average*			
				NMI	ARI	F1	DLD	NMI	ARI	F1	DLD	NMI	ARI	F1	DLD	NMI	ARI	F1	DLD	NMI	ARI	F1	DLD
Baselines	CISIR [Jiang *et al.*, 2018]			20.47	6.45	12.92	25.01	65.77	32.89	35.46	47.11	46.62	3.37	20.60	27.17	64.33	45.57	40.32	48.95	49.30	22.07	27.33	37.06
	PtrNet [Yu and Joty, 2020]			21.05	8.45	13.74	20.13	68.02	31.59	30.76	45.31	60.53	37.14	**44.20**	54.22	71.36	51.10	46.92	48.99	55.24	32.07	33.91	42.16
	DialBERT [Li *et al.*, 2020]			25.57	10.97	20.13	40.45	71.65	40.05	38.67	47.56	54.61	8.15	16.49	39.30	15.46	11.37	30.29	21.74	41.82	17.64	26.40	37.26
	SSE2E [Liu *et al.*, 2020]			35.75	25.45	22.13	41.52	73.16	42.80	40.45	49.66	62.61	20.58	18.20	41.52	35.20	25.12	27.88	34.50	51.68	28.49	27.17	41.80
	CATD [Tan *et al.*, 2019]			36.46	24.13	23.04	41.39	74.15	43.21	44.70	50.35	65.85	47.14	30.03	52.30	70.46	51.01	48.57	51.65	61.73	41.37	36.59	48.92
	DAG-LSTM [Pappadopulo *et al.*, 2021]			34.97	25.16	24.25	43.95	73.87	40.67	44.25	51.65	66.27	45.37	31.32	46.59	76.94	51.64	45.35	50.97	63.01	40.71	36.29	48.29
MUIDIAL	Intent	U2UDIAL	U2TDIAL	**39.64**	**28.99**	**32.17**	**52.42**	**76.97**	**44.35**	**45.62**	**57.46**	**72.45**	**52.31**	38.65	**57.91**	**79.25**	**56.52**	**49.37**	**61.25**	**67.08**	**45.54**	**41.45**	**57.26**
Variants		✓	✓	37.46	27.56	29.60	49.67	73.95	42.79	42.14	54.40	66.06	48.02	33.05	51.95	72.94	54.71	47.14	53.16	62.60	43.27	37.98	52.30
	✓	✓		22.71	17.45	16.24	25.13	67.95	39.56	39.51	49.75	66.13	43.31	35.51	54.51	73.38	53.30	47.05	50.07	57.54	38.41	34.58	44.87
	✓		✓	36.25	25.08	24.07	44.17	73.24	41.85	39.57	50.15	63.54	31.15	21.16	43.76	46.61	36.36	38.52	35.65	54.91	33.61	30.83	43.43

Table 2: MUIDIAL comparison results of metric NMI, ARI, F1 and DLD (%) on IM→Gitter. **Bold face**: The highest result of each column.

Dataset	IM	Reddit	IRC	Gitter
Domain	social/movie	social/news	tech./Ubuntu	tech./software
#utterance	612,053	126,641	47,394	22,107
#dialogue	56,562	4,575	2,527	6,315
#speaker	5,352	3,169	4,470	1,217
train/valid/test				
↪ Baselines	9,875/2,010/2,010	3,100/200/210	1,980/134/100	2,100/300/137
↪ MUIDIAL	2,120/190/2,010	1,100/125/210	1,980/134/100	2,100/300/137

Table 3: Details of benchmark datasets. The train/valid/test datasets are partitioned by the benchmark datasets.

Results \ Models	Disentanglement Results	
	Dial. 1	*Dial. 2*
CISIR	(u_1), u_2, (u_4), u_5, u_9, u_{10}	u_3, u_6, (u_7), (u_8)
PtrNet	(u_1), u_2, (u_4), u_5, ~~u_7~~, u_9, u_{10}	u_3, u_6, (u_7), (u_8)
DialBERT	u_1, u_2, u_4, (u_5), ~~u_6~~, ~~u_7~~, (u_9), u_{10}	u_3, (u_6), (u_7), (u_8)
SSE2E	u_1, u_2, u_4, (u_5), ~~u_6~~, ~~u_7~~, (u_9), (u_{10})	u_3, ~~u_4~~, (u_6), (u_7), (u_8), ~~u_{10}~~
CATD	u_1, u_2, ~~u_3~~, (u_4), u_5, u_9, u_{10}	(u_3), ~~u_4~~, u_6, u_7, u_8
DAG-LSTM	u_1, u_2, (u_4), (u_5), u_9, u_{10}	u_3, ~~u_4~~, ~~u_5~~, u_6, u_7, u_8
MUIDIAL	u_1, u_2, u_4, u_5, u_9, u_{10}	u_3, u_6, u_7, u_8
Ground-Truth	u_1, u_2, u_4, u_5, u_9, u_{10}	u_3, u_6, u_7, u_8

Table 4: Case study on Figure 1. ~~u_t~~: Error-disentangled utterances. (u_i): Missing utterances.

better metrics on dialogue disentanglement against other metrics [Jiang *et al.*, 2021], thus we use DLD for tuning hyperparameters in the experiments. The final set of the hyperparameters is shown as follows. The utterance embedding vector and the heuristic feature vector are 768-dimensional. The LSTM hidden size is 256. When training MUIDIAL, the mini-batch size is set to 16. Adam optimizer [Kingma and Ba, 2015] is used to optimize the parameters with the initial learning rate of 5e-4. The experiment environment is a Windows 10 desktop computer with NVIDIA GeForce RTX 2060 GPU, Intel Core i7 CPU, and 32GB RAM.

3.2 Experiment Results

Comparison with SOTA

Baselines. 1) U2U baselines: CISIR is a Siamese hierarchical CNN that estimates the similarity between utterances. PtrNet applies the pointer module on calculating heuristic similarity scores between the current utterance and its former utterances. 2) U2T baselines: DialBERT applies BERT to encode the utterance and predicts the belonging relationship with a BiLSTM classifier. SSE2E utilizes the session-state encoder and "build+update" state-transition model to predict the session-based probability. 3) Hybrid baselines: CATD combines a link-based and a cluster-based model together by weighting the two prediction probabilities. DAG-LSTM enriches the utterance feature embedding with thread encoding and predicts the dialogues with N-ary Tree-LSTM.

Results. Table 2 demonstrates the baseline comparison results. When comparing with the best performing baselines, MUIDIAL achieves the best performance on all four metrics on average, improving by 4.07% (NMI), 4.17% (ARI), 4.86% (F1), and 8.34% (DLD). MUIDIAL also outperforms the baselines on the majority of the individual benchmark datasets. We believe that the performance advantage of MUIDIAL mainly comes from two perspectives: 1) MUIDIAL embeds utterances with multiple aspects, such as intent embedding and heuristic features. Thus it is able to accurately capture the semantics of utterances; 2) MUIDIAL incorporates mutual learning to better take the advantages from both U2U and U2T, thus enhances the prediction accuracy.

Case Study. To qualitatively compare MUIDIAL and baselines, we illustrate how they disentangle the example chat slice presented in Figure 1. The disentanglement results are shown in in Table 4. We can see that the U2U baselines (CISIR and PtrNet) are capable of finding utterances with similar semantic features, but are likely to miss some coupling utterances in the scope of the dialogue (e.g., missing u_1 and u_4 in Dial. 1, and missing u_7 and u_8 in Dial. 2). U2T baselines (DialBERT and SSE2E) are more fitted to detect comprehensive utterances in the dialogue-thread scope, but are likely to miss some semantically similar utterances posted far from the majority (e.g., missing u_9 in Dial. 1). Hybrid baselines (CATD and DAG-LSTM) are likely to make less mistakes than other baselines. Overall, MUIDIAL provides the most accurate disentanglement result.

Component Analysis

Variants. 1) U2UDIAL+U2TDIAL is the MUIDIAL without the intent embedding. 2) Int.+U2UDIAL refers to the models without mutual learning, which use only U2U prediction for disentanglement. 3) Int.+U2TDIAL refers to the models without mutual learning, which use only U2T prediction for disentanglement.

Results. Table 2 presents the performances of MUIDIAL and its variants. We can see that MUIDIAL outperforms all its variants on average. When comparing with MUIDIAL and U2UDIAL+U2TDIAL, removing the intent embedding leads to a moderate decrease of 4.48% (NMI), 2.27% (ARI), 3.47% (F1), and 4.96% (DLD). When comparing MUIDIAL with Int.+U2UDIAL and Int.+U2TDIAL respectively, removing the mutual learning framework leads to a dramatic decrease, where using Int.+U2UDIAL decreases 9.54% (NMI), 7.14% (F1), 6.87% (F1), 12.40% (DLD), and using Int.+U2TDIAL decreases 12.17% (NMI), 11.93% (ARI), 10.62% (F1), 13.83% (DLD). In summary, both the intent-

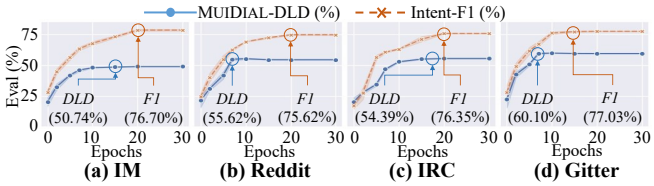

Figure 4: Convergence analysis of intent classification during training of MUIDIAL.

based utterance embedding and mutual-based joint training adopted by MUIDIAL are helpful for dialogue disentanglement, where mutual-based joint training provides the most significant contribution to the effectiveness of MUIDIAL.

Performance of Intent Classification. Figure 4 shows the performance of intent classification during the mutual learning (Epoch 0-30), along with the performance of MUIDIAL. We can see that the intent-F1 scores can reach 75.62%-77.03% at the end of training on all the four datasets, indicating the intent classification performance is relatively satisfactory. We can also see that MUIDIAL converges faster than intent classification. Since MUIDIAL achieves the highest performance when the intent-F1 scores are around 55%, we believe that the current intent classification is "good enough", and using a better intent classification may not upgrade the MUIDIAL performance any more.

Hyper-parameter Analysis

We conduct experiments to investigate the sensitivity of our method to the three hyper-parameters: intent loss weight α, mutual loss weight β, and number of state sub-vectors K. The results are illustrated in Figure 5. For α and β, we first independently train MUIDIAL by varying α from 0 to 1.0. Our model achieves the highest performance when $\alpha = 0.4$ on average. Then, we fix α to 0.4 and vary β from 0 to 0.6. The best performance is achieved at $\beta = 0.3$ for IM and Gitter, and $\beta = 0.4$ for Reddit and IRC. Finally, we vary K from 2 to 7. During the process, we observe that with the increase of K, the value of DLD shows a positive correlation, but the growth is slight.

4 Related Work

Dialogue Disentanglement Models. Apart from the SOTA approaches introduced in Section 4.1, there are some other studies on dialogue disentanglement. Shen et al. [2006] proposed Weighted-SP with three variations of a single-pass clustering algorithm to predict whether the utterance-pair had any potential semantic relationships. Elsner et al. [2010] chose three heuristic features to represent the utterances and trained the utterance-pair prediction by using max-entropy classifiers. DeepQA[Severyn and Moschitti, 2015] and ABCNN[Yin *et al.*, 2016] adopted neural models to predict the QA structure of utterances and obtained independent dialogues. Mehri et al. [2017] utilized RNN as the classifier to predict the "reply-to" links.

Mutual Learning Application. Although mutual learning has not been commonly facilitated in dialogue disentanglement, it is found successful in several other research areas.

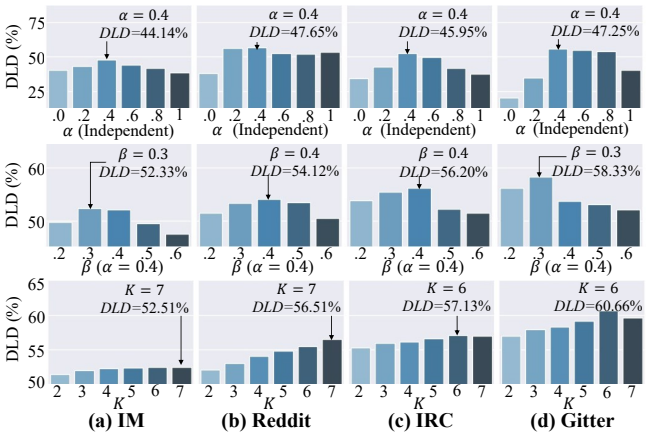

Figure 5: Analysis of hyper-parameters.

Zhang et al. [2018] proposed a deep mutual learning (DML) approach to learn collaboratively in a group of student models throughout the training process, which improved the image detection task on the CIFAR-100 dataset. Wu et al. [2019] utilized a mutual learning module (MLM) to better leverage the correlation of multiple tasks and significantly improved saliency detection accuracy. Liu et al. [2019] proposed the DAML models, which utilized mutual-based attention to predict user rating for item recommendation. Yang et al. [2020] proposed MutualNet, which integrated a mutual learning scheme with both network width and resolution in order to reduce computation resources of perception tasks.

5 Conclusion and Future Work

In this paper, we propose a novel intent-based dialogue disentanglement model MUIDIAL using the mutual learning framework, which enriches the utterance embedding with user intents. A new U2U prediction U2UDIAL and a new U2T prediction U2TDIAL are built separately to predict the "reply-to" and "belong-to" probability matrices. The thread-based transformation (TBT) is then introduced to transform the probability matrices to the same dimensions. Mutual learning framework is used to train U2UDIAL and U2TDIAL. The evaluations on four benchmark datasets show that our model outperforms the baselines by 5% on average.

In the future, we plan to conduct ablation studies on MUIDIAL via utilizing other user-intent encoders. Additionally, we look to introduce some keyword analysis on text fragments such as codes and external links into MUIDIAL, which may further improve the performances of our dialogue disentanglement model.

Acknowledgments

This work is supported by the National Key R&D Program of China under Grant No. 2018YFB1403400, the National Science Foundation of China under Grant No. 61802374, 62002348, and 62072442, and Youth Innovation Promotion Association CAS.

References

[Chen and Manning, 2014] Danqi Chen and Christopher D. Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750, 2014.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

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

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[Jiang *et al.*, 2018] Jyun-Yu Jiang, Francine Chen, Yan-Ying Chen, and Wei Wang. Learning to disentangle interleaved conversational threads with a siamese hierarchical network and similarity ranking. In *NAACL-HLT*, pages 1812–1822, June 2018.

[Jiang *et al.*, 2021] Ziyou Jiang, Lin Shi, Celia Chen, Jun Hu, and Qing Wang. Dialogue disentanglement in software engineering: How far are we? In *IJCAI*, pages 3822–3828, 8 2021.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Kummerfeld *et al.*, 2019] Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros Polymenakos, and Walter S. Lasecki. A large-scale corpus for conversation disentanglement. In *ACL*, pages 3846–3856, July 2019.

[Li *et al.*, 2020] Tianda Li, Jia-Chen Gu, Xiaodan Zhu, Quan Liu, Zhen-Hua Ling, Zhiming Su, and Si Wei. Dialbert: A hierarchical pre-trained model for conversation disentanglement. *arXiv preprint arXiv:2004.03760*, 2020.

[Liu *et al.*, 2019] Donghua Liu, Jing Li, Bo Du, Jun Chang, and Rong Gao. DAML: dual attention mutual learning between ratings and reviews for item recommendation. In *KDD*, pages 344–352, 2019.

[Liu *et al.*, 2020] Hui Liu, Zhan Shi, Jia-Chen Gu, Quan Liu, Si Wei, and Xiaodan Zhu. End-to-end transition-based online dialogue disentanglement. In *IJCAI*, pages 3868–3874, 7 2020.

[Liu *et al.*, 2021] Hui Liu, Zhan Shi, and Xiaodan Zhu. Unsupervised conversation disentanglement through co-training. In *EMNLP*, pages 2345–2356, 2021.

[Mehri and Carenini, 2017] Shikib Mehri and Giuseppe Carenini. Chat disentanglement: Identifying semantic reply relationships with random forests and recurrent neural networks. In *IJCNLP*, pages 615–623, 2017.

[Nguyen *et al.*, 2020] Thanh-Tung Nguyen, Xuan-Phi Nguyen, Shafiq R. Joty, and Xiaoli Li. Efficient constituency parsing by pointing. In *ACL*, pages 3284–3294, 2020.

[Pappadopulo *et al.*, 2021] Duccio Pappadopulo, Lisa Bauer, Marco Farina, Ozan Irsoy, and Mohit Bansal. Disentangling online chats with dag-structured lstms. In **SEM*, pages 152–159, 2021.

[Qu *et al.*, 2018] Chen Qu, Liu Yang, W. Bruce Croft, Johanne R. Trippas, Yongfeng Zhang, and Minghui Qiu. Analyzing and characterizing user intent in information-seeking conversations. In *SIGIR*, pages 989–992, 2018.

[Santos and Embrechts, 2009] Jorge M. Santos and Mark J. Embrechts. On the use of the adjusted rand index as a metric for evaluating supervised classification. In *ICANN*, volume 5769, pages 175–184, 2009.

[Severyn and Moschitti, 2015] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR*, pages 373–382, 2015.

[Shen *et al.*, 2006] Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. Thread detection in dynamic text message streams. In *SIGIR*, pages 35–42, 2006.

[Strehl and Ghosh, 2002] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2002.

[Tan *et al.*, 2019] Ming Tan, Dakuo Wang, Yupeng Gao, Haoyu Wang, Saloni Potdar, Xiaoxiao Guo, Shiyu Chang, and Mo Yu. Context-aware conversation thread detection in multi-party chat. In *EMNLP-IJCNLP*, pages 6455–6460, 2019.

[Wu *et al.*, 2019] Runmin Wu, Mengyang Feng, Wenlong Guan, Dong Wang, Huchuan Lu, and Errui Ding. A mutual learning method for salient object detection with intertwined multi-supervision. In *CVPR*, pages 8150–8159, 2019.

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *NAACL*, pages 1480–1489, 2016.

[Yang *et al.*, 2020] Taojiannan Yang, Sijie Zhu, Chen Chen, Shen Yan, Mi Zhang, and Andrew R. Willis. Mutualnet: Adaptive convnet via mutual learning from network width and resolution. In *ECCV*, volume 12346 of *Lecture Notes in Computer Science*, pages 299–315, 2020.

[Yin *et al.*, 2016] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. ABCNN: attention-based convolutional neural network for modeling sentence pairs. *Trans. Assoc. Comput. Linguistics*, 4:259–272, 2016.

[Yu and Joty, 2020] Tao Yu and Shafiq R. Joty. Online conversation disentanglement with pointer networks. In *EMNLP*, pages 6321–6330, 2020.

[Zhang *et al.*, 2018] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, pages 4320–4328, 2018.

[Zhu *et al.*, 2021] Rongxin Zhu, Jey Han Lau, and Jianzhong Qi. Findings on conversation disentanglement. *arXiv preprint arXiv:2112.05346*, 2021.