Structural Characterization for Dialogue Disentanglement

Anonymous ACL submission

Abstract

Tangled multi-party dialogue contexts lead to challenges for dialogue reading comprehension, where multiple dialogue threads flow simultaneously within a common dialogue history, increasing difficulties in understanding a dialogue history for both human and machine. Previous studies mainly focus on utterance encoding methods with carefully designed features and pay inadequate attention to characteristic features of the structure of dialogues. We specially take dialogue structure factors into account and design a novel model for dialogue disentangling. Based on the fact that dialogues are constructed on successive participation of speakers and interactions between users of interest, we extract clues of speaker property and reference of users to model structural information of dialogues. The proposed method achieves new state-of-the-art on benchmark dataset and contributes to dialogue-related comprehension.

1 Introduction

Communication between multiple parties happens anytime and anywhere, especially as the booming social network services hugely facilitates open discussion, such as group chatting and forum discussion, producing various tangled dialogue logs (Lowe et al., 2015; Zhang et al., 2018; Choi et al., 2018; Reddy et al., 2019; Li et al., 2020a). Whereas, it can be challenging for a new participant to understand the previous chatting log since multi-party dialogues always exhibit disorder and complication (Shen et al., 2006; Elsner and Charniak, 2010; Jiang et al., 2018; Kummerfeld et al., 2019). In fact, it is because of the distributed and random organization, multi-party dialogues are much less coherent or consistent than plain texts. As the example shown in figure 1, the development of a multi-party dialogue has characteristic factors: 1) Random users successively participate in the dialogue and follow certain topics that they are interested in, motivating the development of those topics. 2) Users reply to former related utterances and mention involved users, forming dependencies among utterances. As a result, multiple ongoing conversation threads develop as the dialogue proceeding, which breaks the consistency and hinders both human and machine from understanding context, let alone giving a proper response (Jiang et al., 2018; Kummerfeld et al., 2019; Joty et al., 2019; Jiang et al., 2021). In a word, the behavior of speakers determines the structure of a dialogue passage, and the structure causes problems of reading comprehension.

Figure 1: An example piece of multi-party chatting logs from Ubuntu IRC (Kummerfeld et al., 2019).

Structural features of dialogue context deserves special attention for disentanglement. Disentangling passages or clustering conversation threads contributes to screening concerned parts among contexts, therefore is naturally required by passage comprehension and related downstream dialogue tasks (Elsner and Charniak, 2010; Jia et al., 2020; Liu et al., 2021a), such as response selection, question-answering, etc.

Nevertheless, existing works on dialogue disentanglement remain to be improved (Zhu et al., 2020; Yu and Joty, 2020; Li et al., 2020b), which ignore or pay little attention to characters of dialogues and show suboptimal performance. Earlier works mainly depend on feature engineering (Kummerfeld et al., 2019; Elsner and Charniak, 2010; Yu and Joty, 2020), and use well-constructed handcrafted features to train a naive classifier (Elsner and Charniak, 2010) or linear feed-forward network (Kummerfeld et al., 2019). Recent works are mostly based on two strategies: 1) two-step (Mehri and Carenini, 2017; Zhu et al., 2020; Yu and Joty, 2020; Li et al., 2020b; Liu et al., 2021a) and 2) end-to-end (Tan et al., 2019; Liu et al., 2020). In the two-step method, the disentanglement task is divided into *matching* and *clustering*, which means firstly matching utterance pairs to detect reply-to relations and then dividing utterances into clusters according to the matching score. In the end-to-end strategy, alternatively, for each conversation thread, the state of dialogue is modeled, which is mapped with a new utterance and accordingly updated. At the same time, the new utterance is divided into the best-matched thread. Nonetheless, the essence of both strategies is to model the relations of utterance pairs.

Recently, Pre-trained Language Models (PrLMs) (Devlin et al., 2019; an, 2019; Clark et al., 2020) have brought prosperity to downstream natural language processing tasks by providing contextualized backbones, based on which various works have combined contextualized information with features of dialogues for inspiring achievements (Lowe et al., 2015; Li et al., 2020a; Liu et al., 2021b; Jia et al., 2020; Wang et al., 2020). Studies on dialogue disentanglement also get benefit from PrLMs (Li et al., 2020b; Zhu et al., 2020), whereas there is still room for improvement due to their insufficient enhancement from capturing dialogue structure information.

So as to enhance characteristic structural features of tangled multi-party dialogues, we design a new model as a better solution for the dialogue disentanglement. The structure of a multi-party dialogue is based on the actions of speakers according to the natural development of dialogues. Thus we extract 1) speaker property and 2) reference between users to characterize dependencies of utterances, which is taken into consideration to help with the detection of reply-to relationships. Evaluation is conducted on DSTC-8 Ubuntu IRC dataset (Kummerfeld et al., 2019), where our proposed model achieves new state-of-the-art. Further analyses and applications of our model illustrate the advantages and scalability additionally.

2 Background and Related Work

2.1 Pre-trained Language Models

Pre-trained language models (PrLMs) have brought remarkable achievements in a wide range of natural language processing (NLP) tasks, where BERT (Devlin et al., 2019) is one of the most significant and inspiring pioneers. It was pre-trained on the two self-supervised training objectives, Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) (Devlin et al., 2019). Devoted to NLP tasks, PrLMs often work as a contextualized encoder with some task-oriented layers added. For disentanglement on multi-party dialogues, existing models concatenate utterances to feed into subsequent layers and use the contextualized output for detecting relationships of utterances (Zhu et al., 2020; Li et al., 2020b).

2.2 Dialogue-related Machine Reading Comprehension

Dialogue-related machine reading comprehension (MRC) brings challenges on handling the complicated scenarios from multiple speakers and criss-crossed dependencies among utterances (Lowe et al., 2015; Yang and Choi, 2019; Sun et al., 2019; Li et al., 2020a). A dialogue is developed by all involved speakers in a distributed way, where an individual speaker focuses and declares oneself on some of the topics discussed in the conversation or reply to utterances from other related speakers. Therefore, consistency and continuity are broken by tangled reply-to dependencies between non-adjacent utterances (Li et al., 2020a; Jia et al., 2020; Ma et al., 2021; Li et al., 2021), leading to a graph structure that is quite different from the smooth presentation in plain texts.

Recently, numbers of works of dialogue-related MRC have managed to enhance dialogue structural features in order to deal with dialogue passages better (Jia et al., 2020; Ouyang et al., 2021; Ma et al., 2021; Li et al., 2021), which achieve progress compared to methods that were previously proposed for plain texts. This inspiration impacts and promotes a wide range of dialogue-related

MRC tasks such as response selection (Gu et al., 2020; Liu et al., 2021b), question answering (Ma et al., 2021; Li et al., 2021), emotion detection (Hu et al., 2021) and so on.

2.3 Dialogue Disentanglement

Dialogue disentanglement (Elsner and Charniak, 2010), which is also referred as conversation management (Traum et al., 2004) , thread detection (Shen et al., 2006) or thread extraction (Adams, 2008), has been studied for decades, due to the both of significance and difficulty of understanding long multi-party dialogues. Various methods of dialogue disentanglement have been proposed aiming to cluster utterances.

Early works can be summarized as feature encoder and clustering algorithms. Well-designed handcraft features are constructed as input of simple networks that predict whether a pair of utterances are alike or different, and clustering methods are then borrowed for partitioning (Elsner and Charniak, 2010; Jiang et al., 2018). Researches are facilitated by a large-scale, high-quality public dataset, Ubuntu IRC, created by Kummerfeld et al. (2019). And then the application of FeedForward network and pointer network (Vinyals et al., 2015) leads to significant progress, but the improvement still partially relies on handcraft-related features (Kummerfeld et al., 2019; Yu and Joty, 2020). Then the end-to-end strategy is proposed and fills the gap between the two steps (Liu et al., 2020), where dialogue disentanglement is modeled as a dialogue state transition process, and utterances are clustered by mapping with the states of each dialogue thread. Inspired by achievements of pre-trained language models (Devlin et al., 2019; Clark et al., 2020; an, 2019), approaches based on PrLMs that serve as the utterance encoder are recently proposed (Zhu et al., 2020; Gu et al., 2020).

However, attention paid to the characteristics of dialogues seems to be inadequate. Feature engineering-based works represent properties of individual utterances such as time, speakers, and topics with naive handcraft methods, thus ignoring dialogue contexts (Elsner and Charniak, 2010; Kummerfeld et al., 2019). PrLM-based Masked Hierarchical Transformer (Zhu et al., 2020) utilizes the golden conversation structures to operate attentions on related utterances when training models, which results in exposure bias. DialBERT (Li et al., 2020b), a recent architecture including a BERT (Devlin et al., 2019) and an LSTM (Hochreiter and Schmidhuber, 1997), models contextual clues but no dialogue-specific features, and claims a state-of-art performance.

In this work, we propose a new design of model considering structural characteristics of dialogues, based on the fact that dialogues are developed according to the behavior of speakers so as to disentangle a multi-party dialogue context. In detail, we model dialogue structures with two highlights: 1) speaker properties of each utterance, which helps with the understanding of utterances, and 2) interactions of speakers between utterances, which helps with the development of conversation threads. The resulting model is evaluated on Ubuntu IRC dataset (Kummerfeld et al., 2019), achieving state-of-the-art performance. Analysis verifies the effectiveness of the proposed methods. In addition, we apply our model in various advanced dialogue-related MRC tasks, which show the applicability and generality of structural characterization on multi-party dialogues.

3 Methodology

The definition of the dialogue disentanglement task and details of our model are sequentially presented in this section, illustrating how we make efforts for disentanglement task with dialogue structural features.

3.1 Task Formulation

Suppose that we perform disentanglement to a long multi-party dialogue history $\mathbb{D} = \{U_0, U_2, \ldots, U_n\}$, where $\mathbb{D}$ is composed of n utterances. An utterance includes an identity of speaker and a message sent by this user, thus denoted as $U_i = \{s_i, m_i\}$. As several threads are flowing simultaneously within $\mathbb{D}$, we define a set of threads $\mathbb{T} = \{t_0, t_2, \ldots, t_n\}$ as a partition of $\mathbb{D}$, where $t_i = \{U_{i_0}, \ldots, U_{i_k}\}, 0 \leq i_0 \leq n, 0 \leq i_k \leq n$, denoting a thread of the conversation. In this task, we aim to disentangle $\mathbb{D}$ into $\mathbb{T}$. As indicated before, a multi-party dialogue is constructed by successive participation of speakers, who often reply to former utterances of interest. Thus, a dialogue passage can be modeled as a graph structure whose vertices denote utterances and edges denote reply-to relationships between utterances. Therefore, we focus on finding a parent node for each utterance through inference of reply-to relationship, so as to discover edges and then

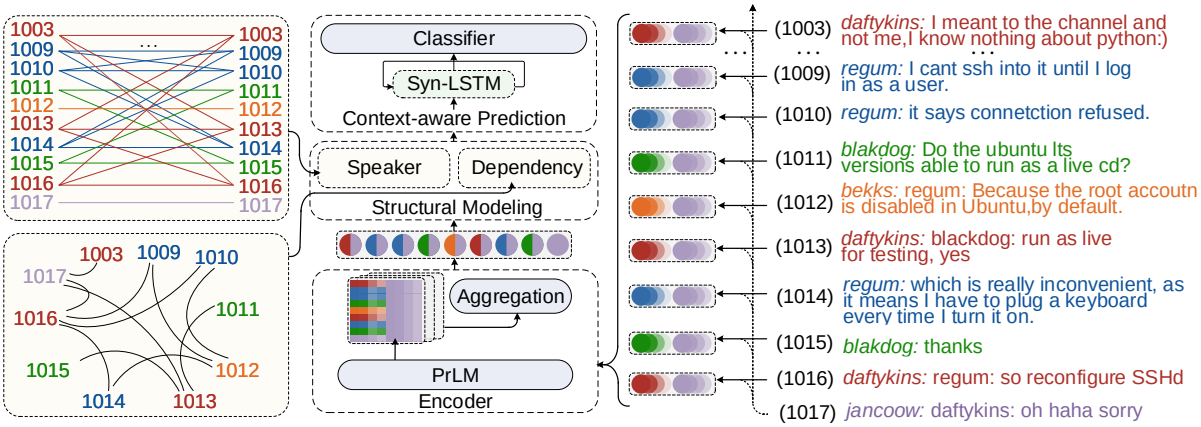

Figure 2: Overview of our model.

determine the graph of a conversation thread.

3.2 Model Architecture

Figure 2 shows the architecture of the proposed model, which is introduced in detail in this part. The model architecture consists of three modules, including text encoder, structural interaction, and context-aware prediction. 1) The utterances from a dialogue history are encoded with a PrLM whose output is then aggregated to context-level in the encoder. 2) The representation is sequentially fed into the structural modeling module, used for dialogue structure features modeling to characterize contexts with speaker-aware and reference-aware features. 3) Then in the prediction module, the model performs a fusion and calculates the prediction of reply-to relationships.

3.2.1 Encoder

Pairwise encoding Following previous works (Zhu et al., 2020; Li et al., 2020b), we utilize a pre-trained language model *e.g.* BERT (Devlin et al., 2019) as an encoder for contextualized representation of tokens. Since chatting records are always long and continuous, it is inappropriate and unrealistic to concatenate the whole context as input. Thus, we concatenate an utterance with each parent candidate separately at the encoder stage, satisfying contextual information from former history.

Assuming that for an utterance U_i, we consider former C utterances (including U_i itself) as candidates for parent node of U_i, the input of a PrLM is in the form of `[CLS]` U_{i-j} `[SEP]` U_i `[SEP]`, where $0 \leq j \leq C-1$. The output is denoted as $H_0 \in \mathbb{R}^{C \times L \times D}$, where C denotes the window length in which former utterances are considered as candidates of the parent, L denotes the input sequence length in tokens, D denotes the dimension of hidden states of the PrLM. Note that there is a situation where the golden parent utterance is beyond the range of $[U_{i-(C-1)}, U_i]$. We label a self-loop for U_i in this case, which means U_i is a beginning of a new dialogue thread as it is too far from the parent, making U_i a root of the thread, which makes sense in the real world, because when users join in a chat (*e.g.* entering a chatting room), they intend to check a limited number of recent messages and make replies, instead of scanning the entire chatting record.

Utterance Aggregation H_0 is pairwise contextualized representations of each pair of the utterance U_i and a candidate U_{i-j}, which will be aggregated to utterance-level representation for further modeling. Since the next sentence prediction information is modeled into the position of `[CLS]`, we simply reserve the representations of `[CLS]`. The concatenated pairwise utterance-level representations from all candidates is denoted as $H_1 \in \mathbb{R}^{C \times D}$, where C denotes the window length and D denotes the dimension of hidden states of the PrLM.

3.2.2 Structural Modeling

Speaker Property Modeling With the goal of enhancing speaker property of each utterance, we follow the mask-based Multi-Head Self-Attention (MHSA) mechanism to emphasis correlations between utterances from the same speaker. The

mask-based MHSA is formulated as follows:

$$\begin{aligned} A(Q, K, V, M) &= \text{softmax}(\frac{QK^T}{\sqrt{d_k}} + M)V, \\ head_t &= A(HW_t^Q, HW_t^K, HW_t^V, M), \\ MHSA(H, M) &= [head_1, , \ldots, head_N]W^O, \end{aligned} \quad (1)$$

where A, $head_t$, Q, K, V, M, N denote the attention, head, query, key, value, mask, and the number of heads, respectively. H denotes the input matrix, and W_t^Q, W_t^K, W_t^V, W^O are parameters. Operator $[\cdot, \cdot]$ denotes concatenation. At this stage, we input the aggregated representation H_1 with a speaker-aware mask:

$$M[i, j] = \begin{cases} 0, & s_i{=}s_j \\ -\infty, & \text{otherwise} \end{cases} \quad (2)$$

$$H_2 = MHSA(H_1, M),$$

where s denotes the speaker identity, M denotes masks of speaker property. The output of MHSA, $H_2 \in \mathbb{C}^{L\times D}$, has the same dimension with H_1. We concatenate H_1 and H_2 and adjust to the same size using a linear layer, resulting in a final output of this module denoted as $H_2 \in \mathbb{C}^{L\times D}$.

Reference Dependency Modeling As discussed above, the relation of references between speakers is the most important and straightforward dependency among utterances, for references indicate interactions between users which is the internal motivation of the development of a dialogue record. To this end, we build a matrix to label the references, which is regarded as an adjacency matrix of a graph representation. In the graph of references, a vertice denotes an utterance and an edge for reference dependence. For example, U_{1012} in Figure 1 mentions and reply to *regum*, forming dependence to utterances from *regum*, i.e., U_{1009}, U_{1010}, and U_{1014}. Thus there are edges from v_{1012} to v_{1009}, v_{1010}, and v_{1014}. Impressed by the activate researches of graph convolutional network (GCN) (Kipf and Welling, 2017), we borrow the relation-modeling of relational graph convolutional network (r-GCN) (Schlichtkrull et al., 2018; Shi and Huang, 2019) in order to enhance the reference dependencies, which can be denoted as follows:

$$h_i^{(l+1)} = \sigma(\sum_{r\in\mathbb{R}}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}),$$

where $\mathbb{R}$ is the set of relationships, which in our module is only reference dependencies. N_i^r denotes the set of neighbours of vertice v_i, which are connected to v_i through relationship r, and $c_{i,r}$ is constant for normalization. $W_r^{(l)}$ and $W_0^{(l)}$ are parameter matrix of layer l. σ is activated function, which in our implementation is ReLU (Glorot et al., 2011; Agarap, 2018). We feed H_2 into this module and derive $H_3 \in \mathbb{C}^{L\times D}$ through dependency modeling of r-GCN.

3.2.3 Context-aware Prediction

We employ a Bi-LSTM (Hochreiter and Schmidhuber, 1997) layer for compensating contextualized information within the whole window of candidates of parent utterances. At the same time, the dialogue structure-aware representation H_3 needs to be combined with the original representation of `[CLS]` H_0 for enhancement.

Motivated by both of them, we employ a Syn-LSTM module (Xu et al., 2021), which was originally proposed for named entity recognition (NER). A Syn-LSTM is distinguished from an additional input gate for an extra source of input, whose parameters are obtained from training, achieving a better fusion of input sources. Thus a layer of Syn-LSTM models the contextual information while the reference dependency is highlighted, enriching relations among parent candidates. The process in a Syn-LSTM cell can be formulated as:

$$\begin{aligned} f_t &= \sigma(\mathrm{W}^{(f)}\mathrm{x}_{1_t} + \mathrm{U}^{(f)}\mathrm{h}_{t-1} + \mathrm{Q}^{(f)}\mathrm{x}_{2_t} + \mathrm{b}_f), \\ o_t &= \sigma(\mathrm{W}^{(0)}\mathrm{x}_{1_t} + \mathrm{U}^{(o)}\mathrm{h}_{t-1} + \mathrm{Q}^{(o)}\mathrm{x}_{2_t} + \mathrm{b}_o), \\ i_{1_t} &= \sigma(\mathrm{W}^{(i)}\mathrm{x}_{1_t} + \mathrm{U}^{(i)}\mathrm{h}_{t-1} + \mathrm{b}_i), \\ i_{2_t} &= \sigma(\mathrm{W}^{(m)}\mathrm{x}_{2_t} + \mathrm{U}^{(m)}\mathrm{h}_{t-1} + \mathrm{b}_m), \\ c_{1_t} &= \tanh(\mathrm{W}^{(k)}\mathrm{x}_{1_t} + \mathrm{U}^{(k)}\mathrm{h}_{t-1} + \mathrm{b}_k), \\ c_{2_t} &= \tanh(\mathrm{W}^{(p)}\mathrm{x}_{2_t} + \mathrm{U}^{(p)}\mathrm{h}_{t-1} + \mathrm{b}_p), \\ c_t &= f_t \odot c_{t-1} + i_{1_t} \odot c_{1_t} + i_{2_t} \odot c_{2_t}, \\ h_t &= o_t \odot \tanh(\mathrm{c}_t), \end{aligned}$$

where x_{1_t} and x_{2_t} are inputs, f_t is a forget gate, o_t is a output gate, i_{1_t} and i_{2_t} are input gates, and W represents parameter. We use the Syn-LSTM in the bi-directional way and the output of it is denoted as $H_4 \in \mathbb{R}^{L\times 2D_r}$, where D_r is the hidden size of the Syn-LSTM.

At this stage, H_4 is the structural feature-enchanced representation of each pair of the utterance U_i and a candidate parent utterance U_{i-j}. To measure the correlations of these pairs, we follow previous work (Li et al., 2020b) to consider the Siamese architecture between each pair of

Model	VI	ARI	1-1	F1	P	R
			Test Set			
FeedForward (Kummerfeld et al., 2019)	91.3	–	75.6	36.2	34.6	38.0
×10 union (Kummerfeld et al., 2019)	86.2	–	62.5	33.4	40.4	28.5
×10 vote (Kummerfeld et al., 2019)	91.5	–	76.0	38.0	36.3	39.7
×10 intersect (Kummerfeld et al., 2019)	69.3	–	26.6	32.1	67.0	21.1
Elsner (Elsner and Charniak, 2008)	82.1	–	51.4	15.5	12.1	21.5
Lowe (Lowe et al., 2017)	80.6	–	53.7	8.9	10.8	7.6
BERT (Li et al., 2020b)	90.8	62.9	75.0	32.5	29.3	36.6
DialBERT (Li et al., 2020b)	92.6	69.6	78.5	44.1	42.3	46.2
+cov (Li et al., 2020b)	93.2	72.8	79.7	44.8	42.1	47.9
+feature (Li et al., 2020b)	92.4	66.6	77.6	42.2	38.8	46.3
+future context (Li et al., 2020b)	92.3	66.3	79.1	42.6	40.0	45.6
Ptr-Net (Yu and Joty, 2020)	92.3	70.2	–	36.0	33.0	38.9
+ Joint train (Yu and Joty, 2020)	93.1	71.3	–	39.7	37.2	42.5
+ Self-link (Yu and Joty, 2020)	93.0	74.3	–	41.5	42.2	44.9
+ Joint train&Self-link (Yu and Joty, 2020)	94.2	**80.1**	-	44.5	44.9	44.2
BERT_{base} (Our baseline)	91.4	60.8	74.4	37.2	34.0	41.2
Our model	**94.6**+3.2	76.8+16	**84.2**+9.8	**51.7**+14.5	**51.8**+17.8	**51.7**+10.5
			Dev Set			
Decom. Atten. (Parikh et al., 2016)	70.3	–	39.8	0.6	0.9	0.7
+feature(Parikh et al., 2016)	87.4	–	66.6	21.1	18.2	25.2
ESIM (Chen et al., 2017)	72.1	–	44.0	1.4	2.2	1.8
+feature (Chen et al., 2017)	87.7	–	65.8	22.6	18.9	28.3
MHT (Zhu et al., 2020)	82.1	–	59.6	8.7	12.6	10.3
+feature (Zhu et al., 2020)	89.8	–	75.4	35.8	32.7	34.2
DialBERT (Li et al., 2020b)	94.1	81.1	85.6	48.0	49.5	46.6
BERT_{base} (Our baseline)	92.8	74.4	80.8	40.8	37.7	42.7
Our model	**94.4**+1.6	**81.8**+7.4	**86.1**+5.3	**52.6**+11.8	**51.0**+13.3	**54.3**+11.6

Table 1: Experimental results on the Ubuntu IRC dataset (Kummerfeld et al., 2019).

$[U_i, U_{i-j}]$ $1 \leq j \leq C-1$ and the pair of $[U_i, U_i]$:

$$\mathrm{H}_{5_j} = [\mathrm{p}_{ii}, \mathrm{p}_{ij}, \mathrm{p}_{ii} \odot p_{ij}, \mathrm{p}_{ii} - \mathrm{p}_{ij}],$$

where p_{ij} is the representation for the pair of $[U_i, U_{i-j}]$ from H_4, and we got $H_4 \in \mathbb{R}^{L \times 8D_r}$. H_5 is then fed into a classifier to predict a parent utterance from all parent candidates. Cross-entropy loss is used as model training object.

4 Experiments

Our proposed model is evaluated on a large-scale multi-party dialogue log dataset Ubuntu IRC (Kummerfeld et al., 2019), which is also used as a dataset of DSTC-8 Track2 Task4. The results show that our model surpasses the baseline significantly and achieves a new state-of-the-art.

4.1 Dataset

Ubuntu IRC (Internet Relay Chat) (Kummerfeld et al., 2019) is the first available dataset and also the largest and most influential benchmark corpus for dialogue disentanglement, which promotes related researches heavily. It is collected from `#Ubuntu` and `#Linux` IRC channels in the form of chatting logs. The usernames of participants are reserved, and reply-to relations are manually annotated in the form of `(parent utterance, son utterance)`. Table 2 shows statics of Ubuntu IRC.

	Passages	Utterances	Links	Avg. users
Train	153	22,0463	69,395	130.3
Dev	10	12,500	2,607	128.1
Test	10	15,000	5,187	156.9

Table 2: Statistics of Ubuntu IRC (Kummerfeld et al., 2019).

4.2 Metrics

Reply-to relations We calculate the accuracy for the prediction of parent utterance, indicating the inference ability for reply-to relations.

Disentanglement For the goal of dialogue disentanglement, threads of a conversation is formed by clustering all related utterances bridged by reply-to relations, in other words, a connected subgraph. At this stage, we use metrics to evaluate following DSTC-8, which are scaled-Variation of Information (VI) (Kummerfeld et al., 2019), Adjusted rand index (ARI) (Hubert and Arabie, 1985), One-to-One Overlap (1-1) (Elsner and

Model	VI	ARI	1-1	F1	P	R
BERT$_{base}$	91.7	74.6	80.2	33.5	32.16	35.0
Ablation study						
+ speaker	94.0	81.2	84.9	45.0	44.7	45.3
+ reference	94.1	82.4	85.6	47.4	47.4	47.4
+ Both	**94.4**	**81.8**	**86.1**	**52.6**	**51.0**	**54.3**
Aggregation methods						
w/ max-pooling	94.1	80.0	85.3	50.8	**52.5**	49.2
w/ [CLS]	**94.4**	**81.8**	**86.1**	**52.6**	51.0	**54.3**
Layers of Syn-LSTM						
w/ 1 layer	**94.4**	**81.8**	**86.1**	**52.6**	51.0	**54.3**
w/ 2 layers	94.0	78.2	84.6	50.4	50.9	50.0
w/ 3 layers	94.3	79.6	85.3	52.2	**51.9**	52.6

Table 3: Results of architecture optimizing experiments.

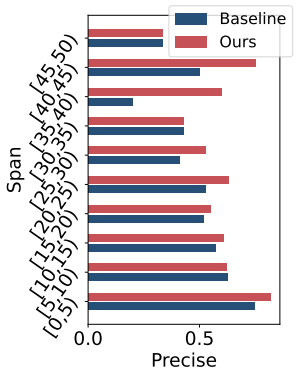

Figure 3: Span length.

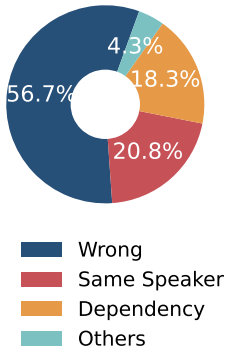

Figure 4: Bad case study.

Charniak, 2010), precision (P), recall (R), and F1 score of clustering. Note that in the table of results, we present 1-VI instead of VI (Kummerfeld et al., 2019), thus we expect larger numerical values for all metrics indicating a stronger performance.

4.3 Setup

Our implementations are based on *Transformers* Library (Wolf et al., 2020). We fine-tune our model employing AdamW (Loshchilov and Hutter, 2019) as the optimizer. The learning rate begins with 4e-6. In addition, due to the trade-off for computing resources, the input sequence length is set to 128, which our inputs are truncated or padded to, and the window width of considered candidates is set to 50.

4.4 Experimental Results

As is presented in Table 1, the experimental results show that our model outperforms all baselines by a large margin as the annotated different values. It is also shown that our model achieves superior performance on most metrics compared to previously proposed models as highlighted in the table, making a new state-of-the-art (SOTA).

5 Analysis

5.1 Architecture Optimizing

5.1.1 Ablation Study

We study the effect of speaker property and reference dependency respectively to verify their specific contribution. We ablate either of the characters and train the model. Results in Table 3 show that both speaker property and reference dependency are non-trivial.

5.1.2 Methods of Aggregation

At the stage of aggregation heading for context-level representations, we consider the influence of different methods of aggregation, i.e., max-pooling and extraction of [CLS] tokens, the models are trained with the same hyper-parameters. Results in Table 3 show [CLS] tokens is a better representation.

5.1.3 Layers of LSTM

To determine the optimal depth of the Bi-Syn-LSTM, we do experiments on the number of layers of a Syn-LSTM, also with the same hyper-parameters. According to the results, as shown in Table 3, we put a one-layer Bi-Syn-LSTM for better performance.

5.2 Prediction Analysis

In order to discuss and intuitively show the progress made by our model, we analyze the predictions from our model and baseline model (i.e., BERT) in different aspects.

1) We catalog reply-to relationships based on the length of their golden spans, and compute the precise of baseline model and ours. Figure 3 shows that our model outperforms baseline by larger margins on links with longer spans (longer than 20 utterances), indicating that our model is more robust on the longer passages.

2) To find out how the speaker property and reference dependency benefit the prediction, we select bad cases of the baseline and study the predictions made by our model on them, as depicted in Figure 4. It is observed that our model corrects 43.3% of the wrong cases from the baseline model, among which 47.9% of the predicted reply-to relations points to utterance pairs with the same speaker annotations, and 42.2% of

them points to pairs with mentioned dependency. As the illustration shows, our model effectively captures the structural information caused by these characters and thus gains improvement.

5.3 Metrics

The used metrics are explained and analyzed briefly for a better understanding of model performance. Details are shown in A.1.

6 Applications

Empirically, it is consistent with our intuition that clarifying the structure of a passage helps with reading comprehension. Accordingly, we assume that disentangling a dialogue passage facilitates reading comprehension. This section aims to verify whether dialogue-related MRC tasks benefit from disentanglement with experiments conducted on different tasks and domains.

6.1 Response Selection

The dataset of DSTC7 subtask1 (Gunasekara et al., 2019) is a benchmark of response selection tasks, derived from Ubuntu chatting logs, which is challenging because of massive scale. As shown in Table 4, it contains hundreds of thousand dialogue passages, and each dialogue has speaker-annotated messages and 100 response candidates.

For implementation, pre-processed context passages are firstly fed into the trained model for disentanglement to obtain predicted partitions of context utterances. Then when dealing with the response selection task, we add a self-attention layer to draw attention between utterances within a common cluster in the hope of labels of clusters leading to better contributions to performance.

6.2 Dialogue MRC

We also make efforts to apply disentanglement on span extraction tasks of question answering datasets, where we consider multi-party dialogue dataset Molweni (Li et al., 2020a), a set of speaker-annotated dialogues with some questions whose answers can be extracted from contexts, which is also collected from Ubuntu chatting logs. Statistics are shown in Table 4. Based on the fact that passages in Molweni are brief compared to other datasets we used, utterances tend to belong to the same conversation session through criss-crossed relations. Thus we alternatively leverage labels of reply-to relations from our model, and build graphs among utterances.

6.3 Open-domain QA

As the former two datasets are both extracted Ubuntu IRC chatting logs, we additionally consider an open-domain dataset, FriendsQA (Yang and Choi, 2019), which contains daily spoken languages from the TV show *Friends*, shown in Table 4. FriendsQA gives QA questions and is coped with in the same way as the Molweni dataset.

	DSTC-7	**Molweni**	**FriendsQA**
Train (dial. / Q)	100,000/–	8,771 / 24,682	973 / 9,791
Dev (dial. / Q)	5000/–	883 / 2,513	113 / 1,189
Test (dial. / Q)	1000/–	100 / 2,871	136 / 1,172
Utterances	3-75	14	173
Responses	100	-	-
Open-domain	N	N	Y

Table 4: Statistics of datasets for applications.

Model	**DSTC-7**		**Molweni**		**FriendsQA**	
	R@1	MRR	EM	F1	EM	F1
Public Baseline	-	-	45.3	58.0	45.2	–
BERT$_{base}$	51.2	60.9	45.7	58.8	45.2	59.6
w/ label	51.4	61.5	46.1	61.7	45.2	60.9

Table 5: Results of application experiments.

Results presented in Table 5 indicate that our model bring consistent profits to downstream tasks. Here we only consider naive baselines and straightforward methods for simplicity and fair comparison, which suggests there is still latent room for performance improvement.

7 Conclusion

In this paper, we study disentanglement on long multi-party dialogue records and propose a new model by paying close attention to the characteristics of dialogue structure, i.e., the speaker property and reference dependency. Our model is evaluated on the largest latest benchmark dataset Ubuntu IRC, where experimental results show the advancement of our method compared to previous work and reach a new SOTA performance. In addition, we analyze the contribution of each structure-related feature by ablation study and the effect of the different model architecture. Our work discloses that speaker property and dependency are significant characters of dialogue contexts and deserve studies in multi-turn dialogue modeling.

References

Holland. Adams, Paige. 2008. Conversation thread extraction and topic detection in text-based chat.

Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *ArXiv preprint*, abs/1803.08375.

Yinhan Liu an. 2019. Roberta: A Robustly Optimized BERT Pretraining Approach. *ArXiv preprint*, abs/1907.11692.

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, Vancouver, Canada. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

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

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

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, Columbus, Ohio. Association for Computational Linguistics.

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

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware BERT for multi-turn response selection in retrieval-based chatbots. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2041–2044. ACM.

Chulaka Gunasekara, Jonathan K. Kummerfeld, Lazaros Polymenakos, and Walter Lasecki. 2019. DSTC7 task 1: Noetic end-to-end response selection. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 60–67, Florence, Italy. Association for Computational Linguistics.

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

Dou Hu, Lingwei Wei, and Xiaoyong Huai. 2021. DialogueCRN: Contextual reasoning networks for emotion recognition in conversations. 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 7042–7052, Online. Association for Computational Linguistics.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

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

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, New Orleans, Louisiana. Association for Computational Linguistics.

Ziyou Jiang, Lin Shi, Celia Chen, Jun Hu, and Qing Wang. 2021. Dialogue disentanglement in software engineering: How far are we? *ArXiv preprint*, abs/2105.08887.

Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Gabriel Murray. 2019. Discourse analysis and its applications. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 12–17, Florence, Italy. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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, Florence, Italy. Association for Computational Linguistics.

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

Jiaqi Li, Ming Liu, Zihao Zheng, Heng Zhang, Bing Qin, Min-Yen Kan, and Ting Liu. 2021. Dadgraph: A discourse-aware dialogue graph neural network for multiparty dialogue machine reading comprehension. *ArXiv preprint*, abs/2104.12377.

Tianda Li, Jia-Chen Gu, Xiaodan Zhu, Quan Liu, Zhen-Hua Ling, Zhiming Su, and Si Wei. 2020b. Dialbert: A hierarchical pre-trained model for conversation disentanglement. *ArXiv preprint*, abs/2004.03760.

Hui Liu, Zhan Shi, Jia-Chen Gu, Quan Liu, Si Wei, and Xiaodan Zhu. 2020. End-to-end transition-based online dialogue disentanglement. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3868–3874. ijcai.org.

Hui Liu, Zhan Shi, and Xiaodan Zhu. 2021a. Unsupervised conversation disentanglement through co-training. *ArXiv preprint*, abs/2109.03199.

Longxiang Liu, Zhuosheng Zhang, Hai Zhao, Xi Zhou, and Xiang Zhou. 2021b. 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)*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic. Association for Computational Linguistics.

Ryan Lowe, Nissan Pow, Iulian Vlad Serban, Laurent Charlin, Chia-Wei Liu, and Joelle Pineau. 2017. Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue & Discourse*, 8(1):31–65.

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2021. Enhanced speaker-aware multi-party multi-turn dialogue comprehension. *ArXiv preprint*, abs/2109.04066.

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, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021. Dialogue graph modeling for conversational machine reading. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3158–3169, Online. Association for Computational Linguistics.

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, Austin, Texas. Association for Computational Linguistics.

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

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.

Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 35–42.

Zhouxing Shi and Minlie Huang. 2019. A deep sequential model for discourse parsing on multi-party dialogues. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7007–7014. AAAI Press.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Ming Tan, Dakuo Wang, Yupeng Gao, Haoyu Wang, Saloni Potdar, Xiaoxiao Guo, Shiyu Chang, and Mo Yu. 2019. Context-aware conversation thread detection in multi-party chat. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6456–6461, Hong Kong, China. Association for Computational Linguistics.

David R. Traum, Susan Robinson, and Jens Stephan. 2004. Evaluation of multi-party virtual reality dialogue interaction. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700.

Weishi Wang, Steven C.H. Hoi, and Shafiq Joty. 2020. Response selection for multi-party conversations with dynamic topic tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6581–6591, Online. Association for Computational Linguistics.

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

Lu Xu, Zhanming Jie, Wei Lu, and Lidong Bing. 2021. Better feature integration for named entity recognition. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3457–3469, Online. Association for Computational Linguistics.

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

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, Online. Association for Computational Linguistics.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3740–3752, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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.

A Appendix

A.1 Metrics

The metrics for evaluating performance of disentanglement are described as follows.

1) scaled-Variation of Information. For the two partition X and Y of set S, $VI(X;Y) = H(X,Y) - I(X,Y)$, where $H(X,Y)$ is the joint entropy of X and Y and $I(X,Y)$ is the mutual information between X and Y, both can be easily calculated from the contingency table. Following previous work(Kummerfeld et al., 2019), VI is scaled to be positive and between 0 and 1. i.e., $1 - VI/log_2(n)$, where n is the number of elements in the set S. Thus a bigger number means the two partitions are more similar.

2) Adjusted Rand Index. The adjusted Rand index is the corrected-for-chance version of the Rand index (Hubert and Arabie, 1985). ARI measures the links between elements under two partitions and indicates how much links lies in the i-th part of the predicted partition X and the j-th part of the ground truth partition Y. Given a contingency table, ARI can be formulated as:

$$\frac{\sum_{ij} C_{n_{ij}}^2 - [\sum_i C_{a_i}^2 \sum_j C_{b_j}^2]/C_{n_{ij}}^2}{\frac{1}{2}[\sum_i C_{a_i}^2 + \sum_j C_{b_j}^2] - [\sum_i C_{a_i}^2 \sum_j C_{b_j}^2]/C_{n_{ij}}^2}$$

, where a_i is the summation if row i and b_j is the summation of column j. C denotes combinatorial number.

3) One-to-One Overlap. One-to-one overlap, also called one-to-one accuracy, is calculated as the percentage overlap by pairing up clusters from two

partitions to maximize overlap using the methods of max-flow algorithm (Elsner and Charniak, 2008), indicating how well a whole conversation can be extracted intact.

4-6) Exact Match. Precise, Recall, and F1 score are used for measuring the exact matching of clusters, where single utterances (clusters only consists of one utterance) is discarded, following previous work.

Recently study made effort to analyze measures (Jiang et al., 2021), where human satisfaction measures are applied on metrics: Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), Shen-F, and F1, results shows that F1 is the most similar to human satisfaction scores, while ARI, NMI and Shen-F seem to tend to overrate disentanglement results but F1 underrates. Here we present a scatterplot 5 based on our experimental results.

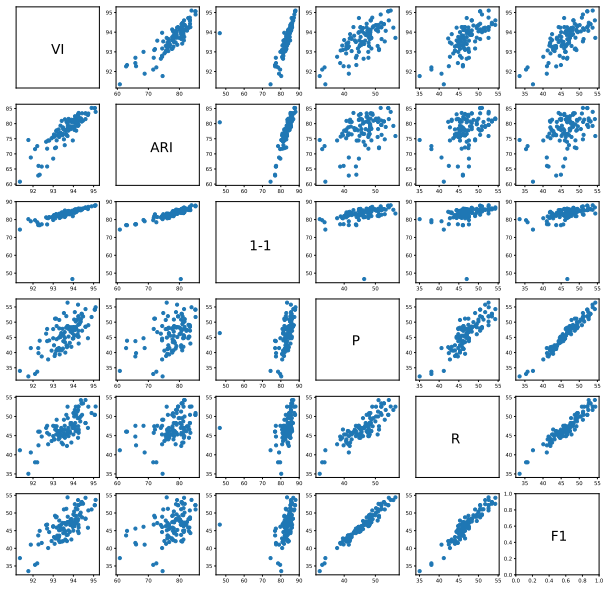

Figure 5: Scatter plots matrix for metrics.