End-to-End Deep Reinforcement Learning for Conversation Disentanglement

Karan Bhukar[1], **Harshit Kumar**[1], **Dinesh Raghu**[1], **Ajay Gupta**[*2]

[1] IBM Research
[2] Meta
karan.bhukar1@ibm.com, harshitk@in.ibm.com, diraghu1@in.ibm.com, guptaajay@fb.com

Abstract

Collaborative Communication platforms (e.g., Slack) support multi-party conversations which contain a large number of messages on shared channels. Multiple conversations intermingle within these messages. The task of conversation disentanglement is to cluster these intermingled messages into conversations. Existing approaches are trained using loss functions that optimize only local decisions, i.e. predicting reply-to links for each message and thereby creating clusters of conversations. In this work, we propose an end-to-end reinforcement learning (RL) approach that directly optimizes a global metric. We observe that using existing global metrics such as variation of information and adjusted rand index as a reward for the RL agent deteriorates its performance. This behaviour is because these metrics completely ignore the reply-to links between messages (local decisions) during reward computation. Therefore, we propose a novel thread-level reward function that captures the global metric without ignoring the local decisions. Through experiments on the Ubuntu IRC dataset, we demonstrate that the proposed RL model improves the performance on both link-level and conversation-level metrics.

Introduction

In recent times, particularly during the global pandemic, there has been an unprecedented proliferation of online communication, on both professional and personal fronts, via online collaboration platforms (Slack), forum discussions (Stackoverflow) and social media (WhatsApp). These communication platforms allow multiple participants to engage in concurrent conversations on shared channels. As a result, it can be quite hard for a reader to understand the context of an on-going chat. The task of conversation disentanglement aims at addressing this issue by identifying separate conversations from a stream of messages. For example, consider a stream of 7 messages in Figure 1. The goal of conversation disentanglement is to identify the 2 conversations (highlighted by red and blue color) along with the reply-to links (indicated by arrows) between the messages in each conversation. Disentangled conversations are readily consumable by downstream tasks, such as response prediction (Kumar, Agarwal, and Joshi 2019; Pandey, Raghu, and Joshi 2020; Tao et al. 2021), dialog act classification(Kumar et al. 2018), question answering(Verma et al. 2019), etc.

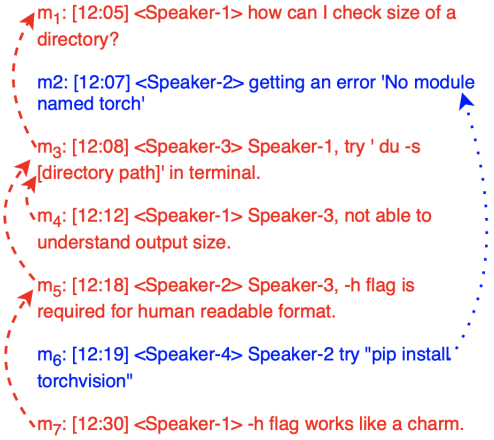

Figure 1: An example of a stream of messages. The two conversations in them are color coded and the reply-to links are illustrated using arrows.

Existing approaches (Ma, Zhang, and Zhao 2022; Yu and Joty 2020; Kummerfeld et al. 2018; Pappadopulo et al. 2021; Li et al. 2020) for conversation disentanglement follow a two-step strategy: (1) for each message predict a parent message to which it is connected by the reply-to link, and (2) infer conversation trees (or clusters) guided by the predicted links from step (1). Existing approaches are trained to optimize the local decisions (i.e., reply-to link prediction) independent of each other and then use global (conversation-level) metrics, such as Adjusted Rand Index (ARI) and Variation of information (VI), for evaluation. While it is desirable to optimize the model on global metrics, it is quite challenging to directly optimize the model in an end-to-end manner because global metrics cannot be computed until all local decisions are finalized.

To address the aforementioned limitation, we propose an end-to-end Reinforcement Learning (RL) based approach that directly optimizes based on a global metric. Our RL policy generates a distribution over all possible reply-to link combinations for all messages in the input message stream.

[*]This work was done while Ajay Gupta was at IBM Research.

The reward is computed by taking into account the reply-to link predictions of all messages, thereby enabling the policy to directly optimize using a global metric. Existing conversation-level metrics for conversation disentanglement represents a conversation tree as a bag of messages. Such representation is devoid of the local link information between messages, therefore using them as input to a reward function explicitly suggests the RL agent to ignore the reply-to linkages during reward computation. To overcome this limitation, we define a novel reward function at the conversation thread-level. This reward function is calculated using the proposed Thread Level FBCubed (TL-FBC) metric, which captures both link-level and conversation-level information between messages, thereby providing better signals to the RL agent.

We evaluate the proposed RL based approach on the widely used Ubuntu IRC dataset (Kummerfeld et al. 2018). We find that our RL based approach that uses our novel TL-FBC metric as reward is significantly better than baselines on both link-level and conversation-level metrics. Overall, our contributions in this paper are:

1. We propose a novel RL based approach[1] for learning the task of conversation disentanglement which can directly optimize global metrics.
2. We propose a novel reward function at a conversation thread-level which captures both link-level and conversation-level information between messages.
3. Experiments on the widely used Ubuntu IRC dataset show that our proposed RL based approach performs significantly better than baselines on both link-level and conversation-level metrics.

Related Work

This section gives a brief overview of the prior works in conversation disentanglement and of the RL approaches used in the similar problem settings. Conversation disentanglement has been a challenging task for a long time, researchers are working on this problem for decades. This task was earlier known as conversation management (Traum, Robinson, and Stephan 2004), thread detection (Shen et al. 2006), and thread extraction (Adams and Martel 2010).

Conversation disentanglement research started with using handcrafted features (Elsner and Charniak 2008; Mehri and Carenini 2017; Jiang et al. 2018) as input to a simple statistical classifier for reply-to link prediction. (Mehri and Carenini 2017) used features from pre-training LSTM on messages to get better representations. (Jiang et al. 2018) used hierarchical Siamese CNN to model similarity between messages in the same conversation. (Kummerfeld et al. 2018) provided a large annotated Ubuntu IRC dataset, which fast-tracked the development of conversation disentanglement approaches. It used glove (Pennington, Socher, and Manning 2014) embeddings and handcrafted features, such as the time difference between messages, speaker id, mention id, etc., to train a naive ensemble of feed-forward classifier. (Pappadopulo et al. 2021) uses DAG-LSTM (Tai, Socher, and Manning 2015) for the systematic inclusion of structured information, such as user turn and mentions, in the learned representation of the conversation context that captures reply-to relation between messages. Similarly (Yu and Joty 2020) uses pointer networks (Vinyals, Fortunato, and Jaitly 2015) to model reply-to link between messages. After the massive success of representations learned from PrLM(Pre-trained Language Models), such as BERT (Devlin et al. 2019), on multiple downstream tasks, the recent works (Zhu et al. 2020; Li et al. 2020; Ma, Zhang, and Zhao 2022) in disentanglement uses BERT to get context based message embeddings for features representation. (Liu, Shi, and Zhu 2021) explored RL in an unsupervised setting for conversation disentanglement, whereas ours is the first work to explore RL in a supervised setting for conversation disentanglement.

Previous works (Elsner and Charniak 2008; Kummerfeld et al. 2018; Yu and Joty 2020; Pappadopulo et al. 2021; Ma, Zhang, and Zhao 2022) treat the task of conversation disentanglement as a two step process, (1) linking step: links messages to parent messages by capturing the reply-to link structure. (2) Clustering step: where messages are grouped together based on the links created between the messages to construct conversation tree. Therefore, current approaches try to optimize the local linking decisions and fail to capture global conversation-level information (e.g., interaction between messages and parents of messages) during training, leading to sub-optimal conversation trees. This work proposes a thread-level metric for reward calculation that optimizes the policy network directly on the global level.

RL has been an elegant way to capture global level information and has shown promising results for similar problem settings, such as co-reference resolution (Fei et al. 2019; Clark and Manning 2016; Yin et al. 2018), relation extraction (Zeng et al. 2018; Qin, Xu, and Wang 2018), and entity extraction (Xiao et al. 2020). Inspired by the success of these works, this is the first work that uses RL to model the task of conversation disentanglement.

Methodology

In this section, we formally define the task of conversation disentanglement followed by a description of our proposed Reinforcement Learning based approach.

Task Definition

Let $\mathbf{m} = \{m_1, m_2, \cdots, m_M\}$ be a set of M messages sent by a group of people on a shared channel. Each message m_i belongs to one of the K ($K \leq M$) conversations $\mathcal{C} = \{C_1, C_2, \cdots, C_K\}$, where each conversation C_k is represented as a tree with messages as nodes and reply-to links as edges. The goal of the task is to disentangle the set of messages M into K conversations with the reply-to links. Here the number of conversations K is unknown.

RL Based Approach

This section describes our proposed RL approach for the task of conversation disentanglement. We first describe the policy which is constructed using *Structural BERT*, the

[1] https://github.com/karan121bhukar/RL-ConvDisentanglement

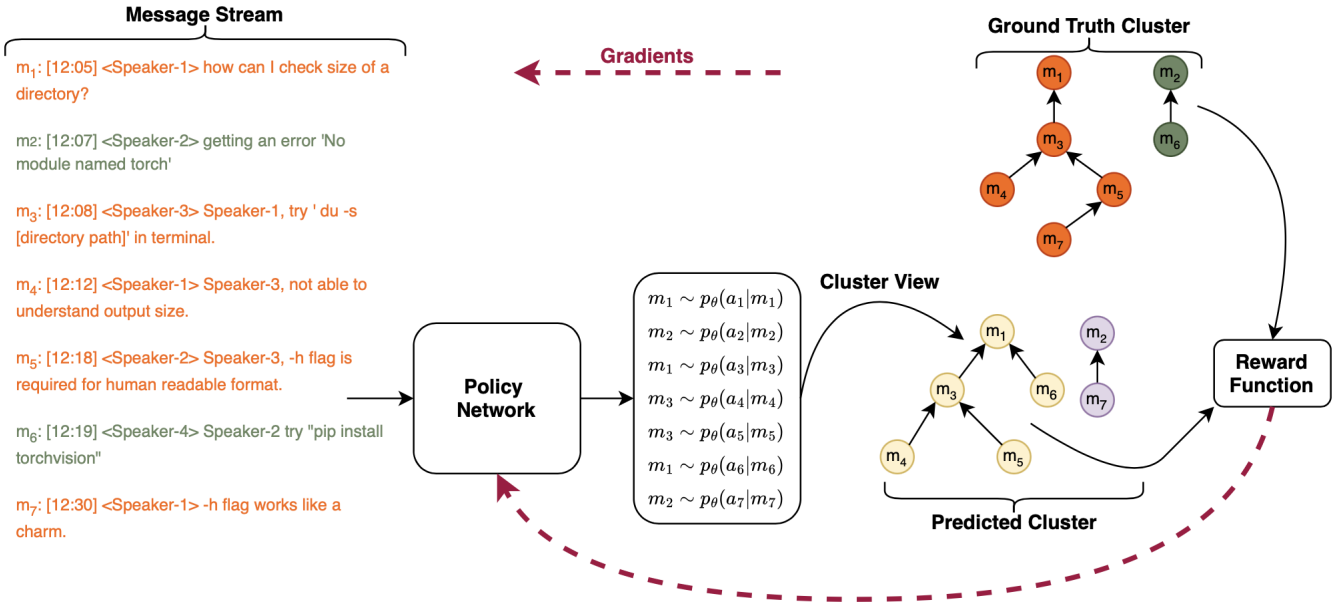

Figure 2: This is the basic framework of our policy gradient method for one action sequence. Policy samples a predicted parent for a message by generating probability over all possible candidate parents. Reward function gives score based on the predicted cluster and red dotted line indicated the gradients based on obtained reward to update parameters of policy

state-of-the-art approach that optimizes only the reply-to link (local) predictions. We then describe the REINFORCE based training algorithm that directly optimizes global metrics (such as ARI or VI) in an end-to-end manner. The overall framework is illustrated in Figure 2.

Given a set of messages $\mathbf{m} = \{m_1, m_2, \cdots, m_M\}$, we predict a set of parent messages $\mathbf{a} = \{a_1, a_2, \cdots, a_M\}$, where $a_i \in \mathbf{m}$ and message m_i is a reply to the message a_i. [2] The policy $\pi_\theta(\mathbf{a}|\mathbf{m})$ outputs a probability distribution over all possible combination of the reply-to link structure, denoted by $\mathcal{A}$. During inference, the conversations can be identified from a set of messages $\mathbf{m}$ by (1) identifying the most likely reply-to link structure $\hat{\mathbf{a}} \sim \text{argmax}_{\mathbf{a}\in\mathcal{A}}\pi_\theta(\mathbf{a}|\mathbf{m})$, and (2) constructing conversation trees by following the predicted reply-to links. Once all the parents $\mathbf{a}$ are identified, the reward $R(\mathbf{a}, \mathcal{C})$ is computed by comparing the predicted conversations to the ground truth conversations. Any global metric can be used as a reward function.

Policy Network: We express the probability of parent messages $\mathbf{a}$ given the messages $\mathbf{m}$ as follows:

$$\pi_\theta(\mathbf{a}|\mathbf{m}) = \prod_{i=1}^{M} p_\theta(a_i|m_i) \quad (1)$$

where $p_\theta(a_i|m_i)$ denotes the distribution over the messages in $\mathbf{m}$ being the parent of the message m_i. Following Kummerfeld et al. (2018), we restrict the candidate parents space to a subset of messages in $\mathbf{m}$. Specifically, we use a fixed window size of w previous messages as candidate parents for m_i, represented as $\mathcal{W}(m_i)$.

[2] If m_i is the first message in the conversation then the reply-to is a self link (i.e., m_i is the parent of m_i).

We use *Structural BERT*, the state-of-the-art approach, that optimizes the link predictions, to compute $p_\theta(a_i|m_i)$. It generates a pairwise representation for each message and a candidate parent pair (m_i, m_j), where $m_j \in \mathcal{W}(m_i)$. These pairwise representations are concatenated and fed to a classifier (Li et al. 2020) to predict a distribution over w candidate parents. The pairwise representation is a combination of a context-aware and a structure-aware representations. The context-aware representation is computed by passing the concatenation of the message and the candidate parent message through BERT and using the [*CLS*] token output. The structure representation is constructed by using a multi headed self attention for encoding correlations between messages from the same speaker and a r-GCN(Schlichtkrull et al. 2018) to encode the relation of references between speakers. The two representations are then combined using a Syn-LSTM (Xu et al. 2021).

Training: The policy network $\pi_\theta(\mathbf{a}|\mathbf{m})$ is trained by maximizing the expected reward $\mathcal{O}_{ER}$ as follows:

$$\mathcal{O}_{ER} = \mathbb{E}_{\mathbf{a}\sim\pi_\theta(\mathbf{a}|\mathbf{m})} R(\mathbf{a}, \mathcal{C}) \quad (2)$$

We use REINFORCE to estimate the gradients of the expected reward by sampling N sets of parents as follows:

$$\nabla_\theta \mathcal{O}_{ER} \approx \frac{1}{N}\sum_{n=1}^{N} \nabla_\theta \log \pi_\theta(\mathbf{a}^n|\mathbf{m})[R(\mathbf{a}^n, \mathcal{C}) - b] \quad (3)$$

$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{M} \nabla_\theta \log p_\theta(a_i^n|m_i)[R(\mathbf{a}^n, \mathcal{C}) - b] \quad (4)$$

where the baseline reward $b = \frac{\sum_{n\in N} R(\mathbf{a}^n, \mathcal{C})}{N}$ is used to reduce the variance in the calculation of approximated gra-

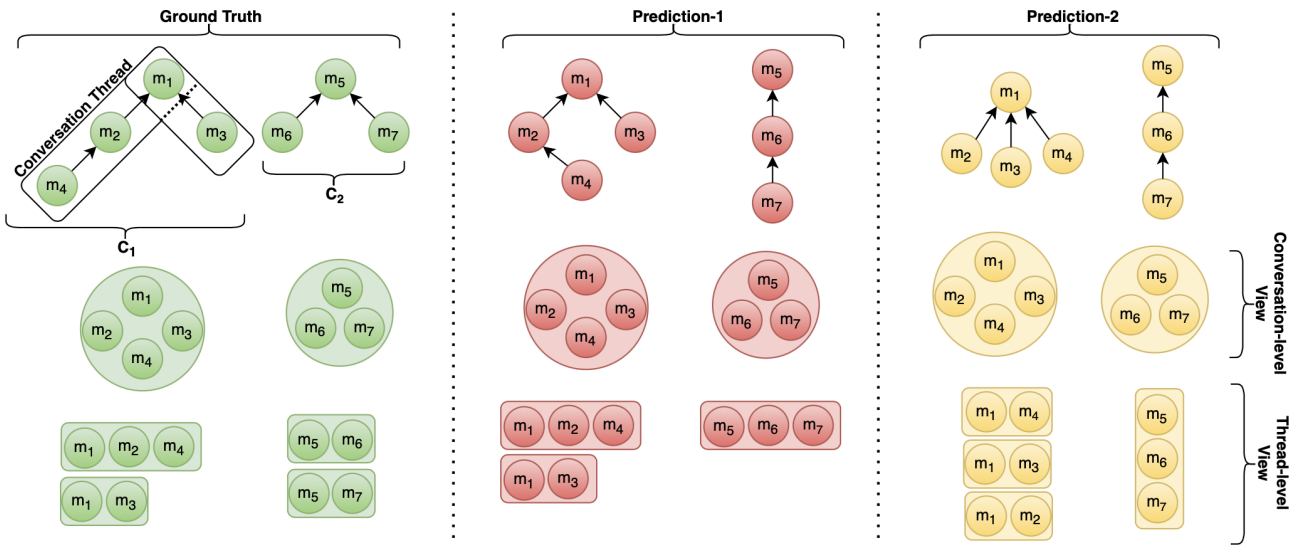

Figure 3: Example contains 2 prediction and a ground truth for 7 messages. ARI, VI rewards for prediction-1 and prediction-2 are 100 for both of them, which is incorrect because both predictions do not match the ground truth. Rewards calculated using micro TL-FBC for prediction-1 and prediction-2 are 93.36 and 87.5, respectively.

dients. Following (Fei et al. 2019), entropy regularization term is added to encourage exploration. In our experiments, the entropy regularization parameter is set to 0.1. Estimating the gradients using Equation 4 ensures (1) each local decision (reply-to link prediction) is guided by the reward that optimizes the global-level metric, and (2) any neural model for conversation disentanglement that optimizes only local decisions can be easily plugged into our RL framework.

Reward Function: We first describe why using existing conversation-level metrics (VI or ARI) as a reward confuses the RL agent, as these metrics lead to a huge space of spurious actions that fetch highest rewards for incorrect actions. We then describe our novel reward function that helps overcome this problem by drastically reducing the space of spurious actions.

Figure 3 describes the issue of using a conversation-level metric as a reward function. The ground truth consists of two conversation trees with 7 messages. And, there are two possible conversation predictions, each containing two conversation trees. Conversation-level metrics such as ARI and VI ignore the reply-to link between the messages within a conversation when computing the metrics. For example, in Figure 3, ARI and VI achieve a perfect score for both the predictions w.r.t the ground truth. Using such a reward metric for our RL setting results in a lot of spurious actions that gets the highest reward even with incorrect reply-to link predictions. These metrics confuse the RL agent by sending incorrect signals. To overcome this issue, we propose a thread-level metric as the reward function which captures both the conversation-level and the link-level information. Our thread-level metric views messages as clusters of conversation threads rather than clusters of conversation trees. For example, the ground truth has four conversation threads ($\{m_1, m_2, m_4\}$, $\{m_1, m_3\}$, $\{m_5, m_6\}$, and $\{m_5, m_7\}$) and two conversation trees (C_1 and C_2). Metrics defined on top of clusters of threads is sensitive to both the global structure and the local link predictions. Constructing clusters of threads results in overlapping clusters (i.e., a single message being a part of multiple threads). For example, m_1 in ground truth belongs to 2 threads, $\{m_1, m_2, m_4\}$ and $\{m_1, m_3\}$. Existing conversation-level clustering metrics such as ARI and VI cannot be directly applied to overlapping clusters(Aroche-Villarruel et al. 2014). So, we use FBcubed (Amigó et al. 2009), a modified version of Bcubed family of metrics(Bagga and Baldwin 1998) suitable for comparing overlapping clusters. The FBcubed metric defined in the original paper is a macro averaged measure. As we use this to compare clusters at a thread-level, we refer to this metric as macro Thread-Level FBcubed (macro TL-FBC).

For the conversation disentanglement task, incorrect reply-to link prediction for a message higher in the conversation tree is much worse than incorrect link prediction for a leaf node. This is because an incorrect upper level link prediction moves the whole sub-tree below it (including itself) to a different cluster; thereby degrading the overall scores for all the involved conversation threads.

To overcome this problem, we propose a micro-averaged version of FBcubed, referred to as micro Thread-Level FBcubed (micro TL-FBC). It assigns weights to message links proportional to the number of threads it belongs to. Let $\mathcal{T}_p$ and $\mathcal{T}_{gt}$ denote the sets of threads in the predicted and ground truth clusters. We represent the subset of predicted threads and subset of ground truth threads which contain messages m_i and m_j as $\mathcal{T}_p(m_i, m_j)$ and $\mathcal{T}_{gt}(m_i, m_j)$ respectively. We compute micro TL-FBC:

$$\text{micro TL-FBC} = \frac{2\Big(\frac{1}{M}\sum_{i=1}^{M} Pr(m_i)\Big)\Big(\frac{1}{M}\sum_{i=1}^{M} Re(m_i)\Big)}{\Big(\frac{1}{M}\sum_{i=1}^{M} Pr(m_i)\Big) + \Big(\frac{1}{M}\sum_{i=1}^{M} Re(m_i)\Big)} \tag{5}$$

$$Pr(m_i) = \frac{\sum\limits_{m_j \in \psi(m_i)} min(|\mathcal{T}_p(m_i, m_j)|, |\mathcal{T}_{gt}(m_i, m_j)|}{\sum_{m_j \in \psi(m_i)} |\mathcal{T}_p(m_i, m_j)|} \tag{6}$$

$$Re(m_i) = \frac{\sum\limits_{m_j \in \phi(m_i)} min(|\mathcal{T}_p(m_i, m_j)|, |\mathcal{T}_{gt}(m_i, m_j)|}{\sum_{m_j \in \phi(m_i)} |\mathcal{T}_{gt}(m_i, m_j)|} \tag{7}$$

where $\psi(m_i)$ and $\phi(m_i)$ are the set of messages that share at least one conversation thread with m_i in $\mathcal{T}_p$ and $\mathcal{T}_{gt}$ respectively.

micro TL-FBC only gets a perfect score, when all the links are predicted correctly. Moreover, at the same time, it also provides the right signal to the RL agent by indicating how close the action is to the global structure. In Figure 3, Prediction-1 has no conversation-level errors and 1 link-level error (m_7). Prediction-2 has no conversation-level errors and 2 link-level errors (m_4 and m_7). Our proposed thread-level metric, micro TL-FBC, gives a score of 93.36 and 87.50 for Prediction-1 and Prediction-2 respectively. This demonstrate two advantages: (1) if the prediction does not have the exact links as in ground truth, its does not get a perfect score, and (2) the score clearly indicates which structure is closer to the ground truth, thereby providing correct signals to the RL agent.

Experimental Setup

Dataset

We use Ubuntu IRC (Internet Relay Chat) (Kummerfeld et al. 2018), the most widely used conversation disentanglement dataset, for our experiments. Out of a total of 220,463 messages spread across 6201 conversations, the number of manually annotated messages are 77,563 – 74,963 from the #Ubuntu IRC channel and 2,600 messages from the #Linux IRC channel). Table 1 presents some statistics of the dataset.

	Train	Dev	Test
# Messages	220,463	12,500	15,000
# Conversations	6201	526	370

Table 1: Statistics of Ubuntu IRC dataset

Evaluation Metrics

Since the conversation disentanglement models first predict links between messages (link prediction) and then use these link information to infer clusters of conversations, we evaluate them using two types of metrics: link-level metrics and conversation-level metrics.

Link-level metrics capture the ability of a model to predict the individual reply-to links between messages. We use precision (P), recall (R) and F1 scores as link-level metrics. Conversation-level metrics capture the ability of a model to extract conversation trees from an entangled message stream. To evaluate performance on a conversation-level, we use the following metrics from the previous works (Kummerfeld et al. 2018; Ma, Zhang, and Zhao 2022; Yu and Joty 2020): scaled-Variation of Information (VI) (Meilă 2007), Adjusted Rand AIndex (ARI) (Hubert and Arabie 1985), precision (P), recall (R) and F1 score.

Baselines

We compare our approach with the following baselines:

1. **Elsner** (Elsner and Charniak 2008): it uses a graph theoretic model with feature set consisting of discourse structures, time gaps between messgaes, and message contents.
2. **Lowe** (Lowe et al. 2017): A rule based model guided by time between messages.
3. **FeedForward** (Kummerfeld et al. 2018): A feed forward network that takes input of hand crafted features such as time difference between messages and users mentioned in a message along word embedding from pre-trained GloVe (Pennington, Socher, and Manning 2014) model.
4. **Ptr-Net** (Yu and Joty 2020): It captures the textual similarity between messages using a pointer network. Input is the similarity value along with the hand crafted features
5. **BERT** (Li et al. 2020): It uses a pre-trained BERT model to build a binary classifier that predicts if a pair of messages have a reply-to link or not.
6. **DialBERT** (Li et al. 2020): Similar to BERT, except BERT is first fine-tuned on a Ubuntu corpus.
7. **DAG-LSTM** (Pappadopulo et al. 2021): This work uses DAG-structured LSTM to incorporate structured information such as user turns and mentions into the conversation context representation.
8. **Structural BERT**[3] (Ma, Zhang, and Zhao 2022):This work incorporates structural features such as speaker identity and mention identity in addition to existing features from Feed Forward model.
9. **Adapted Hierarchical BERT** (Li et al. 2022): It combines the local and global semantics in the context range by encoding each message-pair using BERT and aggregating the chronological context data into the output of BERT using a Bi-LSTM.

Training Details

We implement our RL based approach using Py-Torch (Paszke et al. 2019). AdamW (Loshchilov and Hutter 2017) is used as the optimizer. The set of hyper-parameters that give best results with learning rate and input sequence length

[3]We report the best numbers that we could reproduce using the code available at https://github.com/xbmxb/StructureCharacterization4DD. We incorporated all the suggestions communicated by the authors over email exchanges.

Model	Conversation-Level					Link-Level			Self-Link
	VI	ARI	P	R	F1	P	R	F1	F1
Elsner (Elsner and Charniak 2008)	82.1	-	12.1	21.5	15.5	-	-	-	-
Lowe (Lowe et al. 2017)	80.6	-	10.8	7.6	8.9	-	-	-	-
FeedForward (Kummerfeld et al. 2018)	91.5	-	36.3	39.7	38.0	74.9	72.2	73.5	-
BERT (Li et al. 2020)	90.8	62.9	29.3	36.6	32.5	-	-	-	-
DialBERT (Li et al. 2020)	92.6	69.6	42.3	46.2	44.1	-	-	-	-
Ptr-Net (Yu and Joty 2020)	94.2	**80.1**	44.9	44.2	44.5	74.5	71.7	73.1	91.5
DAG-LSTM (Pappadopulo et al. 2021)	-	-	42.4	41.7	42.0	75.2	72.7	73.9	90.2
Structural BERT (Ma, Zhang, and Zhao 2022)	93.0	68.4	46.7	47.6	47.1	75.3	75.5	75.4	90.1
Adapted Hierarchical BERT (Li et al. 2022)	93.9	76.3	43.3	50.1	46.5	-	-	-	-
Our RL Model	**96.2**	78.5	**51.5**	**52.3**	**51.9**	**83.3**	**83.3**	**83.3**	**92.3**

Table 2: Comparing state-of-the-art models with the proposed RL method on the Ubuntu IRC Dataset

set to 5e-6 and 128 respectively. We set the the number of trajectories N and the candidate parents window size w to 10 and 50 respectively.

Experiments

Our experiments answer the following questions:

1. How does the performance of the proposed RL approach compare to baseline models?
2. How does the novel thread-level metric compare to existing conversation-level metrics when used as a reward?
3. Is the micro TL-FBC better than its macro variant?

Performance Study

Table 2 reports the performance of our RL based approach and the baselines on various link-level and conversation-level metrics. As expected, our model outperforms all baselines on majority of conversation-level metrics. For instance, when compared to the link prediction model used in our policy (i.e., Structural BERT), our RL approach achieves an improvement of 10.2% (46.7 to 51.5), 9.8% (47.6 to 52.3), 10.1% (47.1 to 51.9), 3.4% (93 to 96.2), and 14.7% (68.4 to 78.48) on P, R, $F1$, VI and ARI respectively.

Our RL model greatly outperforms existing models on all three link-level metrics. It achieves a massive improvement of approximately 10% over the previous best performing model. Accuracy of a specific type of links called self-links are crucial for conversation-level metrics. Self links are the reply-to links from messages that point to themselves. A message with self-link indicates that it is the first message of a conversation. Self-link prediction has a cascading effect on the conversation-level metrics, as a wrong self-link prediction results in two conversations being merged together as one conversation, which degrades the conversation-level metrics considerably (Yu and Joty 2020). It can be seen from Table 2 that our RL based approach achieves a two point improvement in self-link prediction compared to Structural BERT, and it can be an aiding factor for the improvement in the conversation-level performance. Consistent improvements in both link-level and conversation-level metrics should be attributed to our thread level reward function micro TL-FBC. By using thread-level clusters rather than conversation-level clusters, we provide positive feedback to the agent even when there is a partial match between the predicted conversation tree and the ground truth. Thus, optimizing using the scores obtained from thread-level metrics moves the policy in the direction which has the potential to improve conversation-level performance. Our thread-level metric (micro TL-FBC) can only achieve a perfect score when all the links in the conversation tree are predicted correctly. This ensures that there are no spurious actions possible which can achieve the same reward as the correct action. Such a reward prevents the RL agent from getting confused leading to improvement in link-level performance.

RewardFunction	Conversation-Level			Link-Level
	VI	ARI	F1	F1
Structural BERT	*93.0*	*68.4*	*47.1*	*75.4*
ARI	94.1	74.3	45.6	72.7
VI	93.6	73.6	44.3	71.6
micro TL-FBC	**96.2**	**78.5**	**51.9**	**83.3**

Table 3: Performance of the RL based approach when trained with conversation-level (ARI and VI) and the proposed thread-level (micro TL-FBC) reward functions.

Conversation-Level Vs Thread-Level Rewards

We perform an ablation to study the effect of different reward functions on our RL model. From the results in Table3, we see that the conversation-level reward functions such as ARI or VI only improves the metric that is being optimized when compared with its non-RL variant (Structural BERT). However, the link-level F1 drops considerably. For example, when ARI is used as a reward function, performance on the ARI improves from 68.4 to 74.3, an improvement of 5.9 absolute points. But the link-level F1 drops from 75.4 to 72.7. Similar behavior is observed for when VI is used as the reward. The main reason for such a behaviour by ARI or VI is because they represent conversations as a bag of nodes and

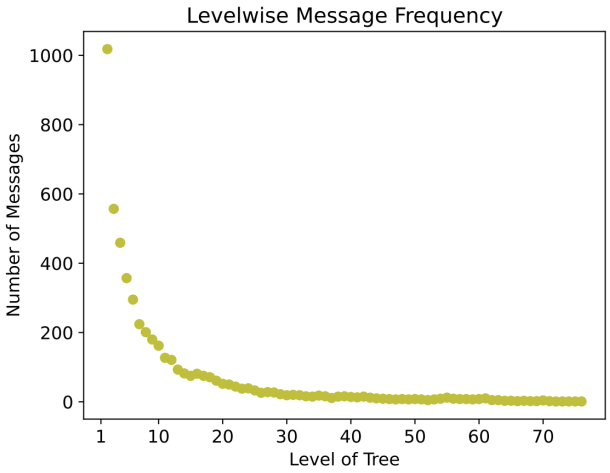

Figure 4: Number of message at different levels of the conversation trees in the Ubuntu IRC dataset.

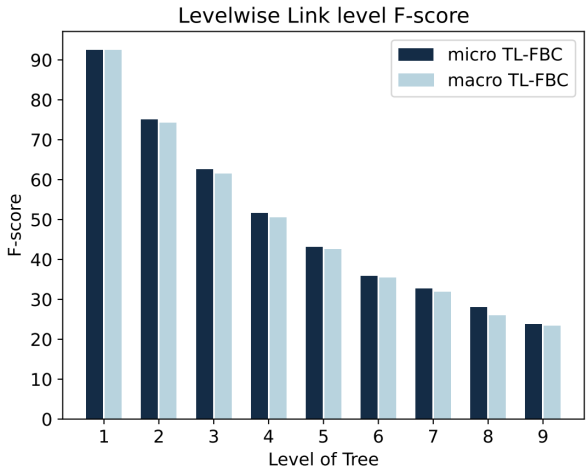

Figure 5: Link-level F1 scores at different levels of the conversation trees.

fail to take into account the link-level information between messages. When micro TL-FBC is used as a reward function, it improves both link-level and conversation-level metrics. By comparing conversation threads instead of conversation trees, micro TL-FBC optimizes both the conversation-level and link-level metrics jointly, resulting in improvements across metrics.

Reward Function	Conversation-Level			Link-Level
	VI	ARI	F1	F1
macro TL-FBC	96.04	77.05	51.00	82.96
micro TL-FBC	**96.23**	**78.47**	**51.86**	**83.34**

Table 4: Performance of the RL approach when using macro TL-FBC and micro TL-FBC as reward function.

Micro vs Macro TL-FBC Rewards

In this sub-section, we compare the macro version of TL-FBC (macro TL-FBC) reward with the proposed micro version of TL-FBC reward (micro TL-FBC). Results in Table 4 shows that the micro TL-FBC reward achieves slightly better performance that the macro TL-FBC. The interesting trend to note is that the improvement, even though marginal, is consistent across both conversation-level and link-level metrics.

Both the macro and the micro variants of TL-FBC compute rewards at a thread-level. The only difference between them is how they weigh the link in the conversation tree during training. Macro assigns equal weights to all links, while micro assigns weights proportional to the number of threads associated with source message. Micro favours message links at higher level in the conversation tree (e.g., a message at the root node) compared to the ones at the lower level in the conversation tree (e.g., a message at the leaf node). Figure 4 shows a histogram of number of messages at each level of the conversation tree in the ground truth. It clearly shows that most of the links are in upper levels. Based on these, we expected that micro version will lead to higher improvements in link predictions at the upper levels as compared to link prediction at lower levels. Figure 5 shows the link-level F1 scores at different levels of predicted conversation tress for both the micro TL-FBC and the macro TL-FBC rewards. Contrary to our aforementioned expectation, it shows that micro TL-FBC has higher F1-scores at all levels, and more so at the lower levels. Inaccurately predicting a reply-to link at higher levels have more damaging effects in comparison to predicting a wrong reply-to link of a message at the lower levels. This is because an incorrect link prediction for a message at an upper level connects affects the whole sub-tree below it (including itself), thereby degrading the overall conversation-level scores for both conversation trees. Thus correct prediction of reply-to links at the higher levels automatically percolates and improves the predictions at the lower levels.

Conclusion

This paper presents an end-to-end reinforcement learning approach for the task of conversation disentanglement which directly optimizes a global metric. We propose a novel thread-level reward function which by representing a conversation as a bag of threads captures both the conversation-level and link-level information. We also propose a micro variant of thread-level FBcubed metric which improves over the macro variant by assigning higher weights to message links at upper levels in the conversation tree. Experiments on the Ubuntu IRC dataset shows that our proposed RL based approach outperforms the existing baselines on both conversation-level and link-level metrics.

References

Adams, P.; and Martel, C. 2010. Conversational thread extraction and topic detection in text-based chat. *Semantic computing*, 87–113.

Amigó, E.; Gonzalo, J.; Artiles, J.; and Verdejo, F. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4): 461–486.

Aroche-Villarruel, A. A.; Carrasco-Ochoa, J. A.; Martínez-Trinidad, J. F.; Olvera-López, J. A.; and Pérez-Suárez, A. 2014. Study of overlapping clustering algorithms based on Kmeans through FBcubed metric. In *Mexican Conference on Pattern Recognition*, 112–121. Springer.

Bagga, A.; and Baldwin, B. 1998. Entity-based cross-document coreferencing using the vector space model. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Clark, K.; and Manning, C. D. 2016. Deep Reinforcement Learning for Mention-Ranking Coreference Models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2256–2262. Austin, Texas: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Elsner, M.; and Charniak, E. 2008. You Talking to Me? A Corpus and Algorithm for Conversation Disentanglement. In *Proceedings of ACL-08: HLT*, 834–842. Columbus, Ohio: Association for Computational Linguistics.

Fei, H.; Li, X.; Li, D.; and Li, P. 2019. End-to-end Deep Reinforcement Learning Based Coreference Resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 660–665. Florence, Italy: Association for Computational Linguistics.

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

Jiang, J.-Y.; Chen, F.; Chen, Y.-Y.; and Wang, W. 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)*, 1812–1822. New Orleans, Louisiana: Association for Computational Linguistics.

Kumar, H.; Agarwal, A.; Dasgupta, R.; and Joshi, S. 2018. Dialogue act sequence labeling using hierarchical encoder with crf. In *Proceedings of the aaai conference on artificial intelligence*, volume 32.

Kumar, H.; Agarwal, A.; and Joshi, S. 2019. A practical dialogue-act-driven conversation model for multi-turn response selection. 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)*, 1980–1989.

Kummerfeld, J. K.; Gouravajhala, S. R.; Peper, J.; Athreya, V.; Gunasekara, C.; Ganhotra, J.; Patel, S. S.; Polymenakos, L.; and Lasecki, W. S. 2018. A large-scale corpus for conversation disentanglement. *arXiv preprint arXiv:1810.11118*.

Li, T.; Gu, J.-C.; Ling, Z.-H.; and Liu, Q. 2022. Conversation-and Tree-Structure Losses for Dialogue Disentanglement. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, 54–64.

Li, T.; Gu, J.-C.; Zhu, X.; Liu, Q.; Ling, Z.-H.; Su, Z.; and Wei, S. 2020. Dialbert: A hierarchical pre-trained model for conversation disentanglement. *arXiv preprint arXiv:2004.03760*.

Liu, H.; Shi, Z.; and Zhu, X. 2021. Unsupervised Conversation Disentanglement through Co-Training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2345–2356. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lowe, R.; Pow, N.; Serban, I. V.; Charlin, L.; Liu, C.-W.; and Pineau, J. 2017. Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue & Discourse*, 8(1): 31–65.

Ma, X.; Zhang, Z.; and Zhao, H. 2022. Structural characterization for dialogue disentanglement. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 285–297.

Mehri, S.; and Carenini, G. 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)*, 615–623. Taipei, Taiwan: Asian Federation of Natural Language Processing.

Meilă, M. 2007. Comparing clusterings—an information based distance. *Journal of multivariate analysis*, 98(5): 873–895.

Pandey, G.; Raghu, D.; and Joshi, S. 2020. Mask & focus: conversation modelling by learning concepts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 8584–8591.

Pappadopulo, D.; Bauer, L.; Farina, M.; Irsoy, O.; and Bansal, M. 2021. Disentangling Online Chats with DAG-Structured LSTMs. *arXiv preprint arXiv:2106.09024*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Qin, P.; Xu, W.; and Wang, W. Y. 2018. Robust distant supervision relation extraction via deep reinforcement learning. *arXiv preprint arXiv:1805.09927*.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Berg, R. v. d.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, 593–607. Springer.

Shen, D.; Yang, Q.; Sun, J.-T.; and Chen, Z. 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*, SIGIR '06, 35–42. New York, NY, USA: Association for Computing Machinery. ISBN 1595933697.

Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1556–1566. Beijing, China: Association for Computational Linguistics.

Tao, C.; Feng, J.; Yan, R.; Wu, W.; and Jiang, D. 2021. A Survey on Response Selection for Retrieval-based Dialogues. In *IJCAI*, 4619–4626.

Traum, D. R.; Robinson, S.; and Stephan, J. 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).

Verma, N.; Sharma, A.; Madan, D.; Contractor, D.; Kumar, H.; and Joshi, S. 2019. Neural conversational qa: Learning to reason vs exploiting patterns. *arXiv preprint arXiv:1909.03759*.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. *Advances in neural information processing systems*, 28.

Xiao, Y.; Tan, C.; Fan, Z.; Xu, Q.; and Zhu, W. 2020. Joint entity and relation extraction with a hybrid transformer and reinforcement learning based model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 9314–9321.

Xu, L.; Jie, Z.; Lu, W.; and Bing, L. 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*, 3457–3469. Online: Association for Computational Linguistics.

Yin, Q.; Zhang, Y.; Zhang, W.-N.; Liu, T.; and Wang, W. Y. 2018. Deep Reinforcement Learning for Chinese Zero Pronoun Resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 569–578. Melbourne, Australia: Association for Computational Linguistics.

Yu, T.; and Joty, S. 2020. Online Conversation Disentanglement with Pointer Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6321–6330. Online: Association for Computational Linguistics.

Zeng, X.; He, S.; Liu, K.; and Zhao, J. 2018. Large scaled relation extraction with reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhu, H.; Nan, F.; Wang, Z.; Nallapati, R.; and Xiang, B. 2020. Who did they respond to? conversation structure modeling using masked hierarchical transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 9741–9748.