# End-to-End Transition-Based Online Dialogue Disentanglement

**Hui Liu**[1] , **Zhan Shi**[1] , **Jia-Chen Gu**[2] , **Quan Liu**[3] , **Si Wei**[3] and **Xiaodan Zhu**[1]
[1]Ingenuity Labs Research Institute & ECE, Queen's University, Canada
[2]University of Science and Technology of China, Hefei, China
[3]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, Hefei, China
{hui.liu, 18zs11, xiaodan.zhu}@queensu.ca, gujc@mail.ustc.edu.cn, {siwei, quanliu}@iflytek.com

## Abstract

Dialogue disentanglement aims to separate intermingled messages into detached sessions. The existing research focuses on two-step architectures, in which a model first retrieves the relationships between two messages and then divides the message stream into separate clusters. Almost all existing work puts significant efforts on selecting features for message-pair classification and clustering, while ignoring the semantic coherence within each session. In this paper, we introduce the first end-to-end transition-based model for online dialogue disentanglement. Our model captures the sequential information of each session as the online algorithm proceeds on processing a dialogue. The coherence in a session is hence modeled when messages are sequentially added into their best-matching sessions. Meanwhile, the research field still lacks data for studying end-to-end dialogue disentanglement, so we construct a large-scale dataset by extracting coherent dialogues from online movie scripts. We evaluate our model on both the dataset we developed and the publicly available Ubuntu *IRC* dataset [Kummerfeld *et al.*, 2019]. The results show that our model significantly outperforms the existing algorithms. Further experiments demonstrate that our model better captures the sequential semantics and obtains more coherent disentangled sessions.[1]

## 1 Introduction

Along with the development of social networks, online group chat channels have embraced a huge success and become more and more popular. The popularization of social APPs, like Slack[2] and Facebook Messenger[3], promote the rapid increase of group conversation messages. When users enter a group chat channel, a bunch of messages will jump out and these messages tend to be related to different topics. These *entangled* messages mix all topics together without illustrating the structure of the conversation, bringing difficulty for users to find topics that they are interested in. Automatic *dialogue disentanglement* will be helpful by separating entangled messages into different sessions and providing users with convenience in finding useful information.

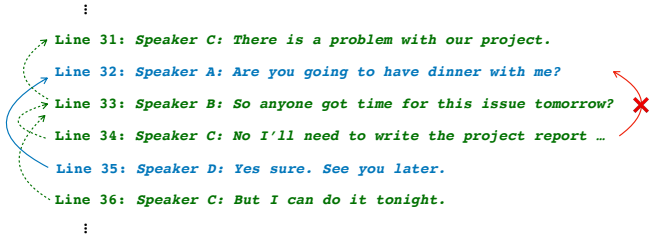



Figure 1: An example of dialogue disentanglement. By considering session history *Line 31*, it is easier for an algorithm to recognize *Line 34* as a response to *Line 33*. If an algorithm only considers message pairs but not session coherence, *Line 34* could be regarded as responding to *Line 32*, resulting in a wrong disentanglement result.

Existing solutions for dialogue disentanglement pay attention to the relationship between two messages. Most of them adopt the two-step architecture: first predicting the relationship between two messages and then separating the message stream into clusters according to the relationship. There are different ways to implement such a two-step architecture. Some early work uses handcrafted features to train a classifier to obtain the global or local coherence of message pairs [Elsner and Charniak, 2010; Elsner and Charniak, 2011]. With the rapid development of deep learning technologies, recent work builds neural models to predict the relationship between two messages [Mehri and Carenini, 2017; Jiang *et al.*, 2018]. The work consider if two messages are in the same session or if one message is replying the other message. Based on the predicted relationship, a clustering algorithm is adopted to separate the messages apart.

There are apparent weaknesses in the two-step methods. In the relationship retrieving step, these methods try to predict the relationships between a message pair but ignore the context in dialogues and the sessions that are already detached. Messages in dialogues tend to be short and simple, which may be orally coherent with many preceding messages without considering the context. An example is given in Figure 1. Simply predicting relationships between message pairs ignores useful session semantics and coherence. In the cluster-

[1]https://github.com/layneins/e2e-dialo-disentanglement
[2]https://slack.com/
[3]https://www.messenger. com/

ing step, existing methods need a meticulously picked clustering algorithm and some require a great deal of human involvement. The weaknesses make such methods less flexible for tackling the disentanglement task.

In order to solve the above problems, we propose the first end-to-end model for online dialogue disentanglement. We formulate dialogue disentanglement as a session state transition problem, where the number of sessions is dynamically maintained. Different from the previous methods, our model captures the sequential information of the dialogue as well as that of the disentangled sessions. When a message is being processed, two actions can be operated on this message: 1) categorize the message into a session that best matches the contextual semantics, or 2) build a new session and initialize the state of the new session with the current message. Through this state transition process, our model focuses on the semantic match between sessions and messages, which shows to be more effective and efficient in our experiments.

To imitate real-life situations, we solve the dialogue disentanglement task in an online manner. In order to overcome the limitations brought by the online training, we propose two learning strategies: teacher-student learning and decision sampling. Both strategies are combined with our model to further improve the performance.

Due to the lack of public benchmarks, we construct a new large-scale dataset from movie scripts for studying dialogue disentanglement. The new dataset contains more than 30,000 intermingled dialogues. We perform experiments on both this newly proposed dataset and the publicly available Ubuntu *IRC* dataset [Kummerfeld *et al.*, 2019]. Results show that our model significantly outperforms the existing two-step methods. Further experiments demonstrate our model's ability in capturing the semantics of the disentangled sessions.

Our main contributions are summarized as follows:

- We propose the first end-to-end model for the online dialogue disentanglement task with two learning strategies. The model captures the sequential information of the disentangled sessions and considers the match of semantics between messages and sessions.
- We release a large-scale dataset for studying dialogue disentanglement. The dataset contains more than 30,000 intermingled dialogues.
- We conduct experiments on two datasets. Results demonstrate that our model significantly outperforms the previous methods by better capturing the semantics of the dialogues and sessions.

## 2 Related Work

Recent work on dialogue disentanglement has mostly adopted a two-step approach: first determining the relationship between two messages, and then separating the messages into clusters. Some previous work predicts if two messages are in the same session. Elsner and Charniak [2010] use handcrafted features to represent a message and use them as the input to train a classifier, and Jiang *et al.* [2018] adopt deep learning methods and use a CNN model as the classifier. Meanwhile, there are some other work targeting at retrieving the "reply-to" relationship between messages. Chen *et al.* [2017] predict the "reply-to" relationship based on text similarity and latent semantic transferability. Guo *et al.* [2018] and Mehri and Carenini [2017] both adopt an RNN to predict if one message is replying to the other message. Apart from building models to extract the relationship between messages, the previous work has also put efforts on designing effective clustering algorithms, including carefully selecting thresholds for the relationship predicted in the first step. Shen *et al.* [2006] run extensive experiments to explore the influence of the threshold used for clustering. Jiang *et al.* [2018] propose a novel similarity ranking method to avoid extensively exploring the setting of the threshold.

There are also some tasks sharing properties in common with dialogue disentanglement such as topic detection and tracking (TDT) [Allan, 2012] and streaming news clustering [Miranda *et al.*, 2018]. However, TDT and streaming news clustering focus on tracking topics, which is very different from dialogue disentanglement. For example, the content in a detached topic is less *sequential* where modelling the coherence of content is less of a concern. The chronological order is less important there (e.g., event locations are among the most prominent features in many top models in TDT). This is very different from dialogue disentanglement, where the disentangled sessions are still conversations and the sequential information in the sessions is critical for the models.

Transition-based methods have been widely used in many sequence prediction tasks [Chen and Manning, 2014; Dyer *et al.*, 2015; Zhang *et al.*, 2016]. Recently, neural-network-based transition models have been studied and achieved good results in different tasks. Our work is the first to propose transition-based neural networks for dialogue disentanglement, which, to our knowledge, has not been investigated.

## 3 Task and Notations

We formulate the task of end-to-end online dialogue disentanglement as a session state transition problem for each message in a dialogue stream. There are two actions for each message: 1) initialize a new session; 2) update an existing session. Since it is an online task, all action decisions are made without knowing the following messages.

The input is a dialogue $\mathcal{D}$ that contains $n$ messages $[u_1, u_2, \ldots, u_n]$. Our goal is to separate the $n$ messages into $K$ sessions $[S^1, S^2, \ldots, S^K]$, where $K$ is unknown to the model. For a given message $u_i$, there exists a session set $\mathbb{S}$ which contains $z(i)$ sessions $[S^1, S^2, ..., S^{z(i)}]$, where $z(i)$ is a function indicating the number of existing sessions when $u_i$ is being processed. Our model needs to decide whether $u_i$ belongs to any session in $\mathbb{S}$. If $u_i$ belongs to $S_j \in \mathbb{S}$, $S_j$ is updated by $u_i$. If $u_i$ does not belong to any existing session, the model will build a new session $S^{z(i)+1}$ and treat $u_i$ as the first message of $S^{z(i)+1}$. Then $S^{z(i)+1}$ is added to $\mathbb{S}$.

## 4 Method

In this section, we first introduce our framework for end-to-end online dialogue disentanglement. Moreover, to eliminate the potential drawback that is caused by the online training process, we propose two learning strategies that can be combined with our model to further improve the performance.

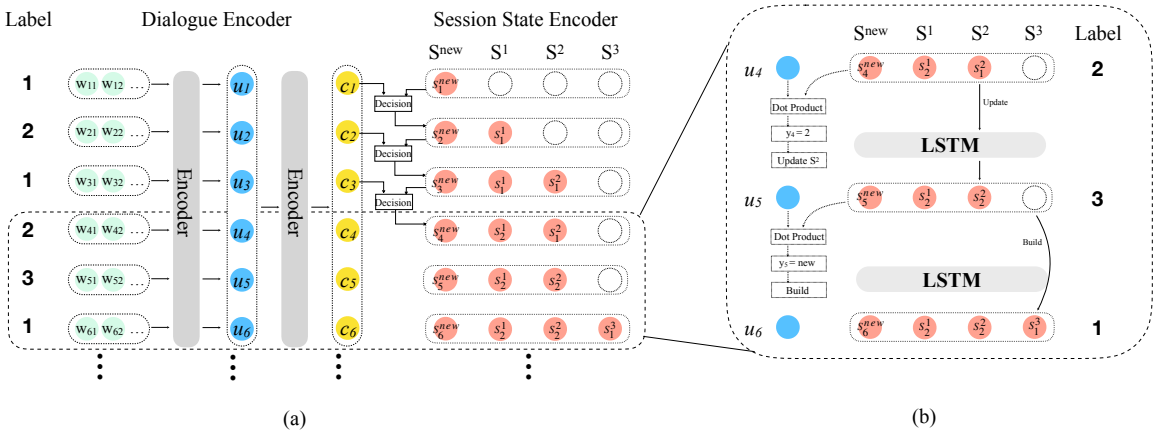



Figure 2: Figure (a) is the overall architecture of our proposed model. Figure (b) is an example from $u_4$ to $u_6$, illustrating how the session state encoder (SSE) updates the state. SSE will make a dot product between $u_4$ and all the elements in the candidate action set, and predict that $u_4$ belongs to $S^2$. Then the state of $S^2$ is updated from $s_1^2$ to $s_2^2$. When $u_5$ is being processed, SSE will predict that $u_5$ belongs to a new session. So the mask of $S^3$ is removed, and we use $u_5$ to initialize the state of $S^3$ as $s_1^3$.

## 4.1 Model

Given a sequence of messages $[u_1, u_2, \ldots, u_n]$, the goal of our model is to learn a probability distribution:

$$P(\mathcal{D}) = \prod_{i=1}^{n} P(y_i \mid y_{<i}, u_{\leq i}, \mathbb{S}) \quad (1)$$

where $y_i$ is the action decision for the message $u_i$.

Our model consists of two parts: the dialogue encoder and session state encoder. The dialogue encoder aims to encode each message into a vector representation. This component captures the semantics of a message and the corresponding preceding context. The second module is a session state encoder that maintains the states of all existing sessions. It should decide if the current message belongs to an existing session or a new session, and update the session state accordingly. Figure 2a presents an overview of our model. We discuss the details of the model in the following sections.

### Dialogue Encoder

The dialogue encoder aims to capture the semantics of a given message and the preceding context. In our model, we use a hierarchical LSTM network [Hochreiter and Schmidhuber, 1997].

For the message-level encoder, given a message $u_i$ that contains $|u_i|$ tokens $\langle w_1, w_2, \ldots, w_{|u_i|}\rangle$, we obtain its vector representation with a LSTM and self-attention (SA) mechanism [Lin *et al.*, 2017] to capture the context information:

$$\langle \boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_{|u_i|}\rangle = \mathrm{LSTM_u}(\langle w_1, w_2, \ldots, w_{|u_i|}\rangle) \quad (2)$$

$$\boldsymbol{u}_i = \mathrm{SA}(\langle \boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_{|u_i|}\rangle) \quad (3)$$

In this way we can obtain all the message representation in a dialogue $\mathcal{D}$ as $\langle \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\rangle$.

For the context-level encoder, we use another LSTM to encoder the context information:

$$\boldsymbol{c}_t = \mathrm{LSTM_c}(\boldsymbol{u}_t, \boldsymbol{c}_{t-1}) \quad (4)$$

We denote the message representations that contain the preceding context information as $\langle \boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n\rangle$.

### Session State Encoder

The key idea behind our model is incrementally building the disentangled sessions according to the action for each message in $\mathcal{D}$. For a given utterance, there will be two actions:

- Build: the current message will be used to initialize a new session, and then the new session is added to $\mathbb{S}$
- Update: the current message will be used to update the state of an existing session in $\mathbb{S}$

Given a message $u_i$, the model will first decide which action to take. Suppose there are $z(i)$ sessions in the session set $\mathbb{S} = [S^1, S^2, \ldots, S^{z(i)}]$. For each session $S^j$, it has a session state representation $\boldsymbol{s}^j$, which contains the sequential information of $S^j$. If the action "Update" is taken, one of the state in $\mathbb{S}$ will be updated. Meanwhile, the number of sessions $K$ is unknown to our model, so there should be a special state indicating when action "Build" should be taken. This state should contain both information in existing sessions and the dialogue history. In our experiment, we calculate the special state as $\boldsymbol{s}_i^{new}$ as:

$$\boldsymbol{s}_i^{new} = \boldsymbol{W_3}[\mathrm{AvgPooling}(\mathbb{S}) : \boldsymbol{c}_{i-1}] \quad (5)$$

where $\boldsymbol{W_3}$ is the parameters. $[:]$ means concatenation and $\mathrm{AvgPooling}(\mathbb{S})$ the average pooling over $\langle \boldsymbol{s}^1, \boldsymbol{s}^2, \ldots, \boldsymbol{s}^{z(i)}\rangle$. In this way, for a given message $u_i$, we have a supplementary session state set as $\langle \boldsymbol{s}_i^{new}, \boldsymbol{s}^1, \boldsymbol{s}^2, \ldots, \boldsymbol{s}^{z(i)}\rangle$.

Then we need to decide which action to take in order to update the states of $\mathbb{S}$. Our model will calculate the dot product between $\boldsymbol{u}_i$ and each state in the supplementary session state set to predict the decision $y_i$:

$$y_i = \mathrm{argmax}\{dot(\boldsymbol{u}_i, \boldsymbol{s}_i^{new}), \max_j dot(\boldsymbol{u}_i, \boldsymbol{s}^j)\} \quad (6)$$

After we obtain the decision $y_i$ for $u_i$, we need to perform state transition on $\mathbb{S}$ according to $y_i$. Our model follows the rule below to conduct the state transition:

- If $y_i = new$, it means our model will take the "Build" action. Then a new session $S^{z(i)+1}$ will be built and $u_i$ will be used to initialize $S^{z(i)+1}$.
- If $y_i = j$, where $j \in [1, 2, \ldots, z(i)]$, our model takes the "Update" action, which means we need to update the state of $S^j$. In order to maintain the sequential information of the session, we use another LSTM as the session encoder to update $S^j$:

$$\boldsymbol{s}^j_{update} = \mathrm{LSTM_s}(\boldsymbol{u}_i, \boldsymbol{s}^j) \tag{7}$$

Notice that $\mathrm{LSTM_s}$ is shared among all the sessions.

An example of the state transition process is shown in Figure 2b.

### 4.2 Training

In this section, we introduce the online training method and several learning strategies to eliminate the drawbacks caused by the training method.

During the online training, the session number is dynamically maintained in our model. In consideration of calculation and convergence, we assume a maximum session number to be $K$[4]. When calculating the loss, we mask all unused session states and calculate a masked cross-entropy loss between the prediction $y_i$ and the ground truth $y_i^\star$:

$$\mathcal{L} = \sum_{\mathcal{D}} \sum_{i=1}^{n} y_i^\star \ln p(y_i) \cdot MASK_{z(i)} \tag{8}$$

$K$ is a hyper-parameter tuned on the development dataset. The setting of $K$ is reasonable for a group chat because there will not be too many sessions simultaneously going on [Aoki *et al.*, 2006]. We will investigate the effect of $K$ in the experiment section.

Since we regard dialogue disentanglement as an online task, two drawbacks are caused due to the limitation of online training:

1. Our model cannot use any future information.
2. There is a discrepancy between how the model is used during training and inference.

In order to overcome the above drawbacks, we propose two training strategies and combine them into our framework: teacher-student learning and decision sampling.

**Teacher-Student Learning**

Teacher-student learning (TSL) is proposed to transfer knowledge from an expert model to a student model [Hinton *et al.*, 2015]. The student model is trained to minimize the difference between its own output distributions and those of the expert model. This approach has shown to be effective in recent studies including speech recognition [Kim *et al.*, 2017] and neural machine translation [Kim and Rush, 2016]. Inspired by these studies, we aim to improve our online model by transferring knowledge from a pretrained offline model.

[4]We use the same assumption for all the baseline experiments. Notice that during inference, when the maximum session number K is reached, action "Build" will not be chosen any more.

Due to the limitation of the online learning strategy, our model cannot see any bidirectional information, which, however, has been proved to be helpful [Peters *et al.*, 2018] for sequential tasks. In our experiments, we change our online model to a bidirectional offline model by replacing $\mathrm{LSTM_u}$, $\mathrm{LSTM_c}$ and $\mathrm{LSTM_s}$ with bidirectional LSTM. We regard the bidirectional offline model as the teacher model. We first pretrain the teacher model and denote its output probability as $P_t(y_i \mid u_{1:n}, \mathbb{S})$. Meanwhile, we regard our model as the student model and denote the output as $P_s(y_i \mid u_{\leq i}, \mathbb{S})$. The goal is to minimize the distance between the output of the teacher model and that of the student model. We calculate Kullback-Leibler (KL) divergence between the two output distributions and the KL loss can be formulated as:

$$\mathcal{L}_{KL} = \sum_{\mathcal{D}} \sum_{i=1}^{n} H(P_t(y_i \mid u_{1:n}, \mathbb{S}), P_s(y_i \mid u_{\leq i}, \mathbb{S})) \tag{9}$$

where $H(P_t, P_s)$ computes cross entropy and the final loss is a combination of the cross-entropy loss and the KL loss:

$$\mathcal{L}_{final} = \alpha\mathcal{L} + (1 - \alpha)\mathcal{L}_{KL} \tag{10}$$

where $\alpha$ is a tunable parameter for balancing the two loss.

**Decision Sampling**

In order to pack the dataset into mini-batch for training, inspired by Chen and Manning [2014], we pre-calculate all the corresponding session states for each message using the ground truth, so the output probability distribution is given by $P(y_i \mid y^\star_{<i})$. However, during inference, ground-truth decisions are unavailable and thus are replaced by decisions generated by the model itself, which will bring an output given by $P(y_i \mid y_{<i})$. This difference yields a discrepancy between how the model is used at training and inference.

Inspired by Scheduled Sampling [Bengio *et al.*, 2015], we similarly propose Decision Sampling (DS) to bridge the gap between training and inference. During training, we add noise to $\sigma\%$ of training data by randomly replacing golden labels. Here $\sigma$ is the sampling ratio which is a hyper-parameter. We do this to imitate the situation during inference when a wrong decision is made. The errors brought by the "fake label" will propagate with the session state updating. We expect our model can still predict the correct labels even when there are some errors in the session states.

## 5 Datasets and Settings

### 5.1 Datasets

There are two datasets used in our experiments: a movie dialogue dataset developed in this paper and the *IRC* dataset proposed by [Kummerfeld *et al.*, 2019].

**Movie Dialogue Dataset**

To contribute to the research on disentanglement, we develop a large-scale dataset from online movie scripts. Messages in the same plot of a movie tend to be coherent. We collect 869 movie scripts that explicitly indicate the plot changing. Then we extract 56,562 sessions from the scripts and manually intermingle these sessions to construct a synthetic dataset. The minimum and maximum session numbers in one dialogue are

2 and 4, respectively. We randomly split the dataset into 29,669/2036/2010 pairs for train/dev/test. We publish our dataset to the research community.

***IRC* Dataset**

The *IRC* dataset contains 153 annotated interleaved dialogues and each of the dialogue consists of 500 messages. It is originally used in the previous study on two-step methods. We separate roughly every 50 continuous messages into a group and obtain 1737/134/104 pairs for train/dev/test, respectively. The difference between the reorganized *IRC* dataset and our proposed dataset is that the minimum and maximum session numbers are 2 and 14 in the reorganized *IRC* dataset.

### 5.2 Training Details

We initialize word embedding using 300-dimension GloVe vectors [Pennington *et al.*, 2014]. Other parameters are initialized by sampling from normal distribution with a standard deviation of 0.1. The mini-batch is 16 and size of hidden vectors in LSTM is 300. We use Adam optimizer [Kingma and Ba, 2014] with an initial learning rate of 5e-4.

## 6 Experiments

### 6.1 Disentanglement Results

For dialogue disentanglement, three clustering metrics are widely used: Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and Shen F value (S-F) [Shen *et al.*, 2006]. ARI is the most strict measure that bases the evaluation on a pairwise basis while NMI penalizes more on the cluster-level. S-F measures how well related messages are grouped, which is a very common and useful metric for dialogue disentanglement. The results are shown in Table 1.[5] On both datasets, our model achieves the best results compared to all previous two-step algorithms, including BERT [Devlin *et al.*, 2019], which has the best performance among all two-step methods. The dialogues in *IRC* contains up to 14 sessions per dialogue, while the maximum session number of our movie dialogue dataset is 4. The results also demonstrate that our model has a good scalability and flexibility. Moreover, the performance of our end-to-end model is further improved when combined with the two learning strategies.

Note that "Oracle" in Table 1 is the optimal performance of two-step methods if all message pair relationships can be correctly predicted. In other words, "Oracle" indicates the effectiveness of the clustering algorithm used in the second step. As we can see, the performance of the clustering algorithm will be a bottleneck of the performance of a two-step method. To achieve good results, two-step methods need both a powerful relationship classifier and a carefully designed clustering algorithm, which yield much inflexibility. For example, choosing to use a different classifier often requires adjusting clustering algorithms. In contrast, our model works in an end-to-end manner, which is more efficient and flexible.

### 6.2 Influence of Session Number

An important metric for evaluating a model is to measure whether the model can disentangle a given dialogue to the correct number of sessions. In the two-step architecture, a wrong threshold for clustering will make the model either keep opening to new sessions or keep categorizing messages to existing sessions. In our experiments, we set a maximum session number $K$ for all the baselines and our model. We study the influence of $K$ by computing the mean squared error (MSE) of the predicted numbers on our movie dialogue dataset. The results are shown in Figure 3. As we can see, our model is better at capturing the semantics of the dialogue because it achieves a lower MSE on session numbers. Meanwhile, a two-step model is more vulnerable to the change of $K$. This is partially due to the threshold used in the clustering step, which has to be adjusted under different experiment settings. In comparison, our model is more flexible when learning automatically to build a new session.

To further test the influence of the session number, we compare the performances of our model and the baseline methods on dialogues that are composed of different numbers of sessions. Since the dialogues in our movie script dataset is composed of 2, 3, and 4 sessions respectively, we use subsets with different session numbers to test the performance, and the results are shown in Figure 4. We can see that the proposed model achieves substantially better results on all the subsets by a rather large margin in comparison to the base-

| Dataset | Movie Dialogue | | | *IRC* | | |
|---|---|---|---|---|---|---|
| Metrics | NMI | ARI | S-F | NMI | ARI | S-F |
| Weighted SP | 18.42 | 4.05 | 52.25 | 25.27 | 2.56 | 33.29 |
| BiLSTM | 22.31 | 8.37 | 55.09 | 47.36 | 1.94 | 40.34 |
| CISIR | 20.47 | 6.45 | 53.77 | 46.62 | 3.37 | 40.78 |
| BERT | 25.57 | 10.97 | 56.91 | 54.61 | 8.15 | 43.87 |
| *Oracle* | 76.19 | 68.13 | 87.24 | 55.25 | 26.92 | 51.91 |
| Our model | 35.3 | 24.9 | 64.7 | 61.4 | 18.0 | 48.19 |
| +TSL | 35.54 | 25.09 | 64.75 | **62.61** | **20.58** | **49.7** |
| +DS | 34.05 | 25.05 | 64.89 | 60.43 | 20.39 | 49.09 |
| +TSL & DS | **35.75** | **25.45** | **64.96** | 61.68 | 19.14 | 48.59 |

Table 1: Performance of dialogue disentanglement on two datasets: the movie dialogues and the IRC dataset. Weighted SP is proposed by [Shen *et al.*, 2006] and CISIR is proposed by [Jiang *et al.*, 2018]. BiLSTM and BERT are both used as the model for retrieving the relationship between a message pair; the clustering algorithm used in the second step is from [Jiang *et al.*, 2018]. *Oracle* indicates the optimal performance of a two-step algorithm if relationships of all message pairs are correctly retrieved in the same dialogue.

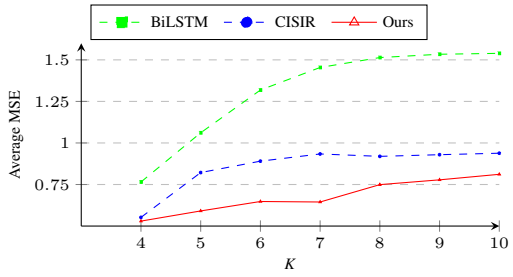



Figure 3: Influence of $K$ on the movie dialogue dataset. The maximum session number is 4.

[5] Significance test $p < 0.05$

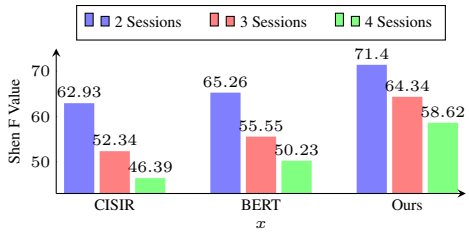
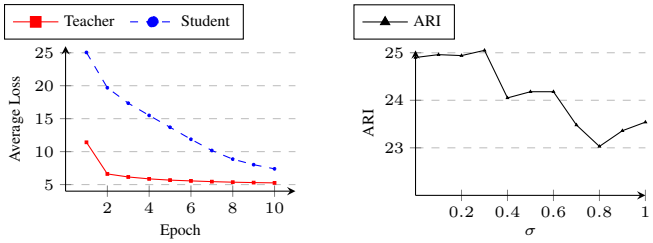



Figure 4: Comparing Shen F Value on movie dialogues with different session numbers.




(a) TSL (b) DS

Figure 5: Experiments on TSL and DS

lines. Meanwhile, for all the methods we compare, the Shen F value will decrease when the dialogues are composed of more sessions, but the degree of drop on our model is more moderate than that of other two-step methods. This also demonstrates our model is more robust than the two-step methods on the change of semantics in dialogues.

### 6.3 Comparison of Learning Strategies

We propose two learning strategies that can further improve the performance of our framework: Teacher-Student Learning (TSL) and Decision Sampling (DS). The two strategies aim to alleviate different drawbacks of the base model.

The goal of TSL is to transfer the knowledge from the teacher model to the student model. In our experiments, we use the bidirectional LSTMs to replace the unidirectional LSTMs and treat this offline model as the teacher model. The assumption is that the teacher model will have a stronger ability to learn the semantics in the dataset. In order to verify the assumption, we compare the training processes of our teacher model and student model in Figure 5a. Notice that we use the same learning rate and optimizer for training the teacher and student. In Figure 5a, we can see that the teacher model converges much faster than the student model, which indicates the teacher has a stronger learning ability.

DS aims to bridge the gap between training and inference. In our experiments, we set a hyper-parameter $\sigma$ to control the sampling ratio. We expect that through randomly replacing some ground-truth labels with wrong labels, we can imitate the inference situation when the model makes a wrong prediction for a given message. Here we explore the influence of $\sigma$. Since all the metrics display a similar fluctuation pattern, we only display the results on ARI in Figure 5b for conciseness. As we can see, a small sampling noise will improve the results and make our model more robust, while when $\sigma$ becomes too large, the performance will be harmed. The results are aligned with theoretical analysis that too much random noise in the dataset can significantly disturb the distribution of the dataset and thus harm the performance.

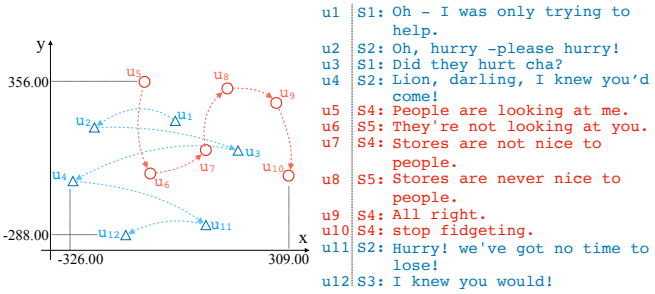



Figure 6: Visualization of the session states

### 6.4 Visualization

One advantage of our model is that it captures the sequential information of the disentangled sessions and categorizes a given message to the session that has the best semantic match. To analyse the update process of sessions, we use t-SNE [Maaten and Hinton, 2008] to map the states of different sessions at different time steps to a 2-dimensional vector. Figure 6 shows an example of how session states change as the dialogue goes on. As we can see, the starting points of sessions are close. With new messages being added, the distance between the states of two sessions becomes larger and the trajectories stretch to different directions, which suggests that our model possesses the capability of distinguishing different meaning between sessions.

## 7 Conclusions

In this paper, we propose the first end-to-end transition-based neural network models for online dialogue disentanglement. Our model captures and utilizes the coherence of sessions that are already disentangled. For each message, our model determines "Build" or "Update" by considering the semantics of the current message, existing sessions, and the dialogue history, and then classifies the message to the best-matched session. Moreover, we propose to enhance our model with two learning strategies. Due to the lack of large-scale datasets, we also develop and contribute a new dataset built on dialogues extracted from movie scripts for dialogue disentanglement research. We conduct experiments on both the movie dialogue dataset and the IRC dataset. The results demonstrate that our model significantly outperforms the baseline methods and can be improved after combined with the proposed learning strategies. Further experiments show that the proposed model is robust and possesses a good ability in capturing the semantics of the disentangled sessions.

## Acknowledgements

The first, second and last author's research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Research Coordinating Committee (CRCC).

## References


[Allan, 2012] James Allan. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media, 2012.

[Aoki *et al.*, 2006] Paul M Aoki, Margaret H Szymanski, Luke Plurkowski, James D Thornton, Allison Woodruff, and Weilie Yi. Where's the party in multi-party?: Analyzing the structure of small-group sociable talk. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 393–402. ACM, 2006.

[Bengio *et al.*, 2015] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

[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.

[Chen *et al.*, 2017] Jun Chen, Chaokun Wang, Heran Lin, Weiping Wang, Zhipeng Cai, and Jianmin Wang. Learning the structures of online asynchronous conversations. In *International Conference on Database Systems for Advanced Applications*, pages 19–34. Springer, 2017.

[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*, pages 4171–4186, 2019.

[Dyer *et al.*, 2015] Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*, 2015.

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

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

[Guo *et al.*, 2018] Gaoyang Guo, Chaokun Wang, Jun Chen, and Pengcheng Ge. Who is answering to whom? finding "reply-to" relations in group chats with long short-term memory networks. In *Proceedings of the 7th International Conference on Emerging Databases*, pages 161–171. Springer, 2018.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 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*, pages 1812–1822, 2018.

[Kim and Rush, 2016] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*, 2016.

[Kim *et al.*, 2017] Suyoun Kim, Michael L Seltzer, Jinyu Li, and Rui Zhao. Improved training for online end-to-end speech recognition systems. *arXiv preprint arXiv:1711.02212*, 2017.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[Lin *et al.*, 2017] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[Mehri and Carenini, 2017] Shikib Mehri and Giuseppe Carenini. 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, 2017.

[Miranda *et al.*, 2018] Sebastião Miranda, Arturs Znotins, Shay B Cohen, and Guntis Barzdins. Multilingual clustering of streaming news. In *EMNLP*, pages 4535–4544, 2018.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Peters *et al.*, 2018] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[Shen *et al.*, 2006] Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 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. ACM, 2006.

[Zhang *et al.*, 2016] Meishan Zhang, Yue Zhang, and Guohong Fu. Transition-based neural word segmentation. In *ACL*, pages 421–431, 2016.