Conversation Disentanglement with Bi-Level Contrastive Learning

Chengyu Huang
National University of Singapore
e0376956@u.nus.edu

Zheng Zhang
Tsinghua University
zhangz.goal@gmail.com

Hao Fei
National University of Singapore
haofei37@nus.edu.sg

Lizi Liao
Singapore Management University
lzliao@smu.edu.sg

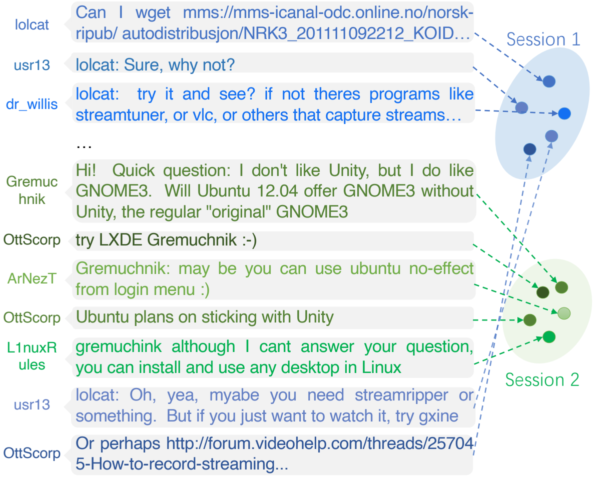
Abstract

Conversation disentanglement aims to group utterances into detached sessions, which is a fundamental task in processing multi-party conversations. Existing methods have two main drawbacks. First, they overemphasize pairwise utterance relations but pay inadequate attention to the utterance-to-context relation modeling. Second, a huge amount of human annotated data is required for training, which is expensive to obtain in practice. To address these issues, we propose a general disentangle model based on bi-level contrastive learning. It brings closer utterances in the same session while encourages each utterance to be near its clustered session prototypes in the representation space. Unlike existing approaches, our disentangle model works in both supervised settings with labeled data and unsupervised settings when no such data is available. The proposed method achieves new state-of-the-art performance results on both settings across several public datasets.

1 Introduction

Multi-party conversations generally involve three or more speakers in a single dialogue, in which the speaker utterances are interleaved, and multiple topics may be discussed concurrently (Aoki et al., 2006). This causes inconvenience for dialogue participant to digest the utterances and respond to a particular topic thread. Conversation disentanglement is the task of separating these entangled utterances into detached sessions, which is a prerequisite of many important downstream tasks such as dialogue information extraction (Fei et al., 2022a,b), state tracking (Zhang et al., 2019; Wu et al., 2022), response generation (Liao et al., 2018, 2021b; Ye et al., 2022a,b), and response ranking (Elsner and Charniak, 2008; Lowe et al., 2017).

There has been substantial work on the conversation disentanglement task. Most of them emphasize on the pairwise relation between utterances in a two-step manner. They predict the relationship between utterance pairs as the first step, followed by clustering utterances into sessions as the second. In the first step, early works (Elsner and Charniak, 2008, 2010) utilize handmade features and discourse cues to predict whether two utterances belong to the same session or whether there is a reply-to relation. The recent development in deep learning inspires the use of neural network such as LSTM or CNN to learn abstract features of utterances in training (Mehri and Carenini, 2017; Jiang et al., 2018). More recently, a number of methods show that BERT in combination with handcrafted features or heuristics remains a strong baseline (Li et al., 2020b; Zhu et al., 2021; Ma et al., 2022). In the second step, the most popular clustering methods use a greedy approach to group utterances by adding pairs (Wang and Oard, 2009; Zhu et al., 2020). There are also some variations incorporating voting mechanism (Kummerfeld et al., 2019), bipartite graph matching (Zhu et al., 2021) or additional tracking models (Wang et al., 2020).



lolcat: Can I wget mms://mms-icanal-odc.online.no/norsk-ripub/ autodistribusjon/NRK3_201111092212_KOID…

usr13: lolcat: Sure, why not?

dr_willis: lolcat: try it and see? if not theres programs like streamtuner, or vlc, or others that capture streams…

…

Gremuchnik: Hi! Quick question: I don't like Unity, but I do like GNOME3. Will Ubuntu 12.04 offer GNOME3 without Unity, the regular "original" GNOME3

OttScorp: try LXDE Gremuchnik :-)

ArNezT: Gremuchnik: may be you can use ubuntu no-effect from login menu :)

OttScorp: Ubuntu plans on sticking with Unity

L1nuxRules: gremuchink although I cant answer your question, you can install and use any desktop in Linux

usr13: lolcat: Oh, yea, myabe you need streamripper or something. But if you just want to watch it, try gxine

OttScorp: Or perhaps http://forum.videohelp.com/threads/257045-How-to-record-streaming...

Figure 1: An example piece of conversation from the Ubuntu IRC corpus. There are distribution patterns in both utterance level and session level.

An obvious drawback of such two-step approach

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 2985–2996
December 7-11, 2022 ©2022 Association for Computational Linguistics

is that the pairwise relation prediction might not capture enough contextual information as the connection between two utterances depends on the contexts in many cases (Liu et al., 2020). Also, focusing on pairwise relations leads to a short-sighted local view. To mitigate this, there are methods trying to introduce additional conversation loss (Li et al., 2020b, 2022) or session classifier (Liu et al., 2021) to group utterances in the same session together. We also see methods leveraging relational graph convolution network (Ma et al., 2022) or masking mechanism in Transformers (Zhu et al., 2020). More directly, end-to-end methods (Tan et al., 2019; Liu et al., 2020) capture the context information contained in detached sessions and calculate the matching degree between a session and an utterance. However, many of such methods are conducted in an online manner which only considers the preceding context. It may lead to biased session representations, introduce noisy utterances to sessions and consequently accumulate errors.

Meanwhile, most of these methods rely heavily upon human-annotated session labels or reply-to relations, which are expensive to obtain in practice. Although there have been a few attempts to tackle this issue, a more general framework that can handle both supervised and unsupervised learning is yet to come. For example, Liu et al. (2021) design a deep co-training scheme with message-pair classifier and session classifier. However, various data augmentation procedures based on heuristics are required for good performance. Chi and Rudnicky (2021) propose a zero-shot disentanglement solution based on a related response selection task. Still, it relies on a closely related dataset that comes from the same Ubuntu IRC source inside DSTC8.

Recently, contrastive learning (Hadsell et al., 2006) has brought prosperity to numbers of machine learning tasks by introducing unsupervised representation learning. Substantial performance gains have been reported in computer vision (He et al., 2020; Chen et al., 2020) and NLP works (Yan et al., 2021; Gao et al., 2021). They believe that good representation should be able to identify semantically close neighbors while distinguishing from non-neighbors. Intuitively, in multi-party conversation, utterances in the same session should semantically resemble each other while be far apart from utterances in other sessions. Instead of handcrafted features such as speaker, mention and time difference *etc*, it provides another option for automatically learn discriminative representations.

In this work, we design a Bi-level Contrastive Learning scheme (Bi-CL) to learn discriminative representations of tangled multi-party dialogue utterances. It not only learns utterance level differences across sessions, but more importantly, it encodes session level structures discovered by clustering into the learned embedding space. Specifically, we introduce session prototypes to represent each session for capturing global dialogue structure and encourage each utterance to be closer to their assigned prototypes. Since the prototypes can be estimated via performing clustering on the utterance representations, it also supports unsupervised conversation disentanglement under an Expectation-Maximization framework. We evaluate the proposed model under both supervised and unsupervised settings across several public datasets. It achieves new state-of-the-art on both.

The contribution is summarized as follows:

- We design a bi-level contrastive learning scheme to learn better utterance level and session level representations for disentanglement.
- We delve into the conversation nature to harvest evidence which supports our model to disentangle dialogues without any supervision.
- Experiments show that the proposed Bi-CL model significantly outperforms several state-of-the-art models both on the supervised and unsupervised settings across datasets.

2 Related Work

2.1 Conversation Disentanglement

Previous methods on conversation disentanglement are mostly performed in a supervised fashion, which can be coarsely organized into two lines: (1) two-step methods which first obtain the pairwise relations among utterances and then disentangle them with a clustering algorithm; and (2) end-to-end approaches which directly assign utterances into different sessions.

The majority of efforts follow the two-step pipeline. Great attention has been devoted to the first step. Early works rely heavily on handcrafted features to represent the utterances for pairwise relation prediction. For example, Elsner and Charniak (2008, 2010) used the speaker, time, mentions, shared word count *etc.* to train a linear classifier for utterance pair coherence. More recent works utilized neural networks to train classifiers. For instance, Mehri and Carenini (2017) and Guo et al.

(2018) leveraged LSTM to predict either the same-session or reply-to probabilities between utterances, while Jiang et al. (2018) combined the output of a hierarchical CNN on utterances with other features to capture the interactions. More recently, Gu et al. (2020) and Li et al. (2020b) used BERT to learn the similarity score in a fixed length context window. For the second step, there has also been progress in exploring an optimal clustering algorithm. Greedy decoding has been a popular choice (Elsner and Charniak, 2010; Jiang et al., 2018). There are also works that train a separate classifier to assign utterance to a thread (Mehri and Carenini, 2017) or design advanced algorithms like bipartite graph matching (Zhu et al., 2021).

On the downside, the pairwise relations, which are predicted typically without considering enough session context, are local and may not reflect how utterances interact in reality. Hence, the clustering step may be undermined subsequently. This motivates end-to-end solutions that aim at assigning the target utterance in each time step with respect to the existing threads or preceding utterances (Liu et al., 2020). Similarly, Yu and Joty (2020) used attention to capture utterance interactions and gradually assign each utterance to its replied-to parent with a pointer module. However, such online manner not only limits the scope of session context but also leads to error accumulation.

There are also studies that work in an unsupervised fashion to avoid the reliance on human-annotation. For example, Liu et al. (2021) designed both message-pair classifier and session classifier to form a co-training algorithm. Chi and Rudnicky (2021) proposed to train a closely-related response selection model for zero-shot disentanglement. The former needs pseudo labeled data to warm-up the training, while the latter gains from training data of the same source. More importantly, a general framework that can handle both supervised and supervised learning is yet to come. In our work, we target at building such a flexible model.

2.2 Contrastive Learning

Contrastive learning learns effective representation by pulling semantically close neighbors together and pushing apart non-neighbors (Hadsell et al., 2006). Recent advances are largely driven by instance discrimination tasks. For example, in the field of computer vision, such methods consist of two key components: image transformation and contrastive loss. The former aims to generate multiple representations about the same image, by data augmentation (Ye et al., 2019; Chen et al., 2020), patch perturbation (Misra and Maaten, 2020), or using momentum features (He et al., 2020). While the latter aims to bring closer samples from the same instance and separate samples from different instances. In the field of natural language processing, contrastive learning has also been widely applied, such as for language model pre-trainining (Yan et al., 2021; Gao et al., 2021).

Despite their improved performance, these instance discrimination methods share a common weakness: the representation is not encouraged to encode the global semantic structure of data (Caron et al., 2020). This is because it treats two samples as a negative pair as long as they are from different instances, regardless of their semantic similarity (Li et al., 2020a). Hence, there are methods which simultaneously conduct contrastive learning at both the instance- and cluster-level (Li et al., 2021; Shen et al., 2021). Likewise, we emphasize leveraging bi-level contrastive objects to learn better utterance level and session level representations.

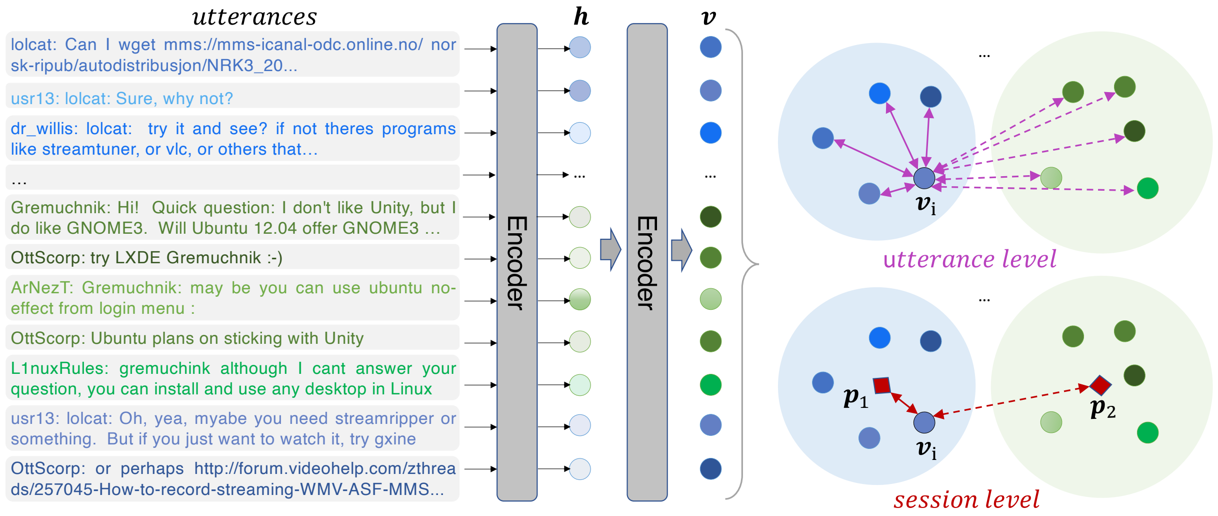
3 Method

The definition of the conversation disentanglement task and details of our model are sequentially presented in this section. Starting from the supervised setting for a clear view, we gradually extend to the unsupervised setting.

3.1 Task Formulation

Given a multi-party conversation history with n utterances $U = \{u_1, u_2, ..., u_n\}$ in chronological order, our goal is to disentangle them into detached sessions $S = \{s_1, s_2, ..., s_k\}$, where each s_i is a non empty subset of U, and S is a partition of U. Each utterance includes an identity of speaker and a message sent by this user.

The task has been popularly formulated as a reply-to relation identification problem to find the parent utterance for every $u_i \in U$. It has also been modeled as sequentially assigning each u_i to already detached sessions in S or create a new session for S. Here, instead of separating local pair and global cluster modeling, we opt for learning more discriminative representations for utterances to push them into different sessions.



lolcat: Can I wget mms://mms-icanal-odc.online.no/ norsk-ripub/autodistribusjon/NRK3_20...

usr13: lolcat: Sure, why not?

dr_willis: lolcat: try it and see? if not theres programs like streamtuner, or vlc, or others that...

...

Gremuchnik: Hi! Quick question: I don't like Unity, but I do like GNOME3. Will Ubuntu 12.04 offer GNOME3 ...

OttScorp: try LXDE Gremuchnik :-)

ArNezT: Gremuchnik: may be you can use ubuntu no-effect from login menu :

OttScorp: Ubuntu plans on sticking with Unity

L1nuxRules: gremuchink although I cant answer your question, you can install and use any desktop in Linux

usr13: lolcat: Oh, yea, myabe you need streamripper or something. But if you just want to watch it, try gxine

OttScorp: or perhaps http://forum.videohelp.com/zthreads/257045-How-to-record-streaming-WMV-ASF-MMS...

Figure 2: Overview of the proposed Bi-CL framework. It incorporates utterance level contrastive loss to discriminate utterances, and uses session level contrastive loss to encourage them flocking around session centers.

3.2 Utterance Encoder

The utterance encoder aims to capture the semantics of a given utterance and its connection to surrounding context. Similar to (Liu et al., 2020), we leverage a hierarchical Bi-LSTM structure similar to (Serban et al., 2017) as illustrated in Figure 2.

For the utterance-level encoder, given each utterance u_i, we tokenize it into tokens $\{t_1, t_2, ..., t_{|u_i|}\}$ and take the Glove embeddings (Pennington et al., 2014). We input these into a bidirectional LSTM and then use a linear transformation with non-linear activation to get the hidden states:

$$\langle \mathbf{h}_1, ..., \mathbf{h}_{|u_i|} \rangle = \delta(\mathbf{W}_1 \cdot BiLSTM(\langle t_1, ..., t_{|u_i|} \rangle)),$$
$$\mathbf{u}_i = SelfAttention(\mathbf{h}_1, \cdots, \mathbf{h}_{|u_i|}),$$

where $\mathbf{W}_1$ is the weight matrix that merges the two direction embeddings of each token, and we use ReLU as the activation δ. We omit the bias term for space limitation. The self-attention mechanism (Lin et al., 2017) is adopted to obtain utterance vectors that represent the overall semantics.

For the context-level encoder, we leverage another bidirectional LSTM to allow utterances to interact with their surroundings and acquire contextual information. Hence, we feed in the local utterance embedding sequence $\langle \mathbf{u}_1, ..., \mathbf{u}_n \rangle$ and obtain the contextual utterance representations $\langle \mathbf{h}'_1, ..., \mathbf{h}'_n \rangle$. It naturally captures information in the utterance itself, in its surrounding utterances and the relative temporal sequence implicitly as:

$$\langle \mathbf{h}'_1, ..., \mathbf{h}'_n \rangle = BiLSTM(\langle \mathbf{u}_1, ..., \mathbf{u}_n \rangle).$$

To further utilize the speaker and mention information of each utterance, we simply concatenate each $\mathbf{h}'_i$ with a padded, multi-hot mention vector $\mathbf{m}_i \in R^{50}$ where the j-th dimension is 1 if the speaker of u_j is the same as that of u_i or mentioned in u_i. This will give the final utterance representations $\langle \mathbf{v}_1, ..., \mathbf{v}_n \rangle$.

3.3 Bi-Level Losses

With encoder network ready at hand, the key is to introduce good objectives for back-propagating the right learning signals. When we have session labels for training data in the supervised learning setting, we aim to train the model so that ideally, (a) utterances in the same ground truth session will be embedded closer while utterances in different sessions will be pulled away; and (b) utterances in each session should be near its session center, or say, prototype. Correspondingly, we introduce utterance-level contrastive loss and session-level contrastive loss to encourage these for learning.

3.3.1 Utterance-level Contrastive Loss

Inspired from the contrastive learning scheme (Khosla et al., 2020) under supervised setting, we contrast an utterance with other utterances in same or different sessions to capture the local structure. Suppose the training dataset $\mathcal{U}$ contains $|\mathcal{U}|$ utterances in total and $y(i)$ denotes the ground truth session assignment of u_i, we define the utterance-level contrastive loss as:

$$L_u = \sum_{i=1}^{|\mathcal{U}|} \frac{-1}{|\mathcal{Y}(i)|} \sum_{j \in \mathcal{Y}(i)} log \frac{exp(\mathbf{v}_i \cdot \mathbf{v}_j / \tau_1)}{\sum\limits_{l \in \mathcal{N}(i,j)} exp(\mathbf{v}_i \cdot \mathbf{v}_l / \tau_1)},$$

where τ_1 is the temperature hyper parameter, $\mathcal{Y}(i)$ contains all the positive utterances that have the same session assignment with u_i, and $\mathcal{N}(i,j)$ contains the set of negative utterances that have different session assignment with u_i, combined with the current positive utterance u_j. Mathematically, we have $\mathcal{Y}(i) \equiv \{j \in \mathcal{U} : y(j) = y(i), j \neq i\}$, and $\mathcal{N}(i,j) \equiv \{l \in \mathcal{U} : y(l) \neq y(i)\} \cup \{j\}$. Ideally, we could use all negative samples as many papers have shown increased performance with increasing number of negatives (He et al., 2020; Henaff, 2020), we set a relatively large number for balancing our computation efficiency.

3.3.2 Session-level Contrastive Loss

In session level, we introduce prototypes to represent each session, and minimize the distance from each utterance to its session prototype while maximize the distances from the utterance to other session prototypes. This incorporates global dialogue semantic structure into the resulting representations. When session labels are available in the supervised setting, suppose $s_i = \{u_1, u_2, ..., u_q\}$, we directly define the prototype p for session s_i:

$$\mathbf{p} = \frac{1}{|q|} \sum_{j=1}^{q} \mathbf{v}_j \ .$$

Therefore, for each conversation U in the training set, we define the session-level contrastive loss:

$$L_s = -\sum_{i=1}^{|U|} \log \frac{exp(\mathbf{v}_i \cdot \mathbf{p}_i / \tau_2)}{\sum\limits_{\mathbf{p}_l} exp(\mathbf{v}_i \cdot \mathbf{p}_l / \tau_2)},$$

where $\mathbf{p}_i$ is the ground truth session prototype for u_i and τ_2 is the temperature hyper parameter.

3.4 Disentangle Sessions

Besides guiding the learning process with bi-level contrastive objects, our disentanglement task naturally involves the session assignment goal. Therefore, the foremost issue is to decide how many sessions the conversation contains. With supervised data, we train a light-weight network to predict K for each conversation. We leverage a two layer feed-forward network enriched with non-linearity. It takes as input the dialogue utterances as well as meta information such as number of speakers n_s and turn number n. The output logits indicate a distribution of the possible K values.

$$\mathbf{d}_U = SelfAttention(\mathbf{v}_1, \cdots, \mathbf{v}_n),$$
$$\mathbf{q} = \delta(\mathbf{W}_2 \cdot \delta(\mathbf{W}_3 \cdot [\mathbf{d}_U;\ n_s;\ n])),$$
$$P(K = k|U) = \frac{\exp(\mathbf{q}_k)}{\sum_{l=1}^{M} \exp(\mathbf{q}_l)},$$

where M is the global maximum session number, and $\mathbf{q} \in \mathbb{R}^M$. We train the network parameters including $\mathbf{W}_2$, $\mathbf{W}_3$ via the K prediction loss:

$$L_k = -\sum_{U \in \mathcal{U}} \log(P(K = \hat{k}|U)),$$

where $\hat{k}$ is the ground truth K for conversation U. In inference, we select the most likely value of K for the K-Means algorithm and constrain $K <= n$.

During training, we also perform K-Means to cluster utterances to mitigate the gap between training and inference. Suppose we obtain a partition $S' = \{s'_1, s'_2, ..., s'_{\hat{k}}\}$ for the conversation U by K-Means, we compute the cluster centroids $\{\mathbf{c}'_1, \mathbf{c}'_2, ..., \mathbf{c}'_{\hat{k}}\}$ by averaging the embeddings of cluster members. We then run Hungarian Algorithm (Kuhn, 1955) to match clusters with sessions, hence align the calculated prototypes with these centroids, *e.g.* $\mathbf{p}_i$ to $\mathbf{c}'_i$. We further introduce a centroid matching loss:

$$L_m = \sum_{U \in \mathcal{U}} \frac{1}{\hat{k}} \sum_{i=1}^{\hat{k}} \|\mathbf{p}_i - \mathbf{c}'_i\|,$$

which ensures that utterance embeddings are clustered according to their ground truth sessions.

To sum up, the final objective for supervised training is as below:

$$L_{supervised} = L_u + \alpha L_s + \beta L_k + \gamma L_m, \quad (1)$$

where α, β, γ are hyper-parameters to adjust the contribution of different factors.

3.5 Unsupervised Extension

In the unsupervised setting, we mainly update the bi-level losses L_u and L_s for representation learning while omit the L_k and L_m losses. In the session level, since we do not know the session labels anymore, we directly estimate the session assignment by clustering utterance embeddings, and then maximize the data log-likelihood. Inspired from (Li et al., 2020a), we perform the two steps iteratively to form an Expectation-Maximization framework. The following shows our objective under the framework. More derivation details can be found in Appendix A.

In a specific iteration, suppose we obtain cluster results as $\{c_1, ..., c_m\}$ by running K-Means on conversation U, maximizing log-likelihood estimation corresponds to finding the utterance encoder network parameters that minimizes the loss:

$$-\sum_{i=1}^{|U|} \log \frac{exp(\mathbf{v}_i \cdot \mathbf{c}_i/\phi_i)}{\sum_{l=1}^{m} exp(\mathbf{v}_i \cdot \mathbf{c}_l/\phi_l)},$$

where ϕ denotes the concentration level of the feature distribution around a cluster centroid c. It encourages utterances to flock around the centroids.

In practice, we cluster the utterances M times with different number of clusters $K = \{k_m\}_{m=1}^{M}$, to achieve a more robust probability estimation of prototypes. Hence the updated session level loss is calculated as:

$$L'_s = -\frac{1}{M} \sum_{i=1}^{|U|} \sum_{m=1}^{M} \log \frac{exp(\mathbf{v}_i \cdot \mathbf{c}_i/\phi_i)}{\sum_{l=1}^{k_m} exp(\mathbf{v}_i \cdot \mathbf{c}_l/\phi_l)},$$

since the number of utterances in conversation U is limited, we set ϕ to a small constant τ'_2.

In the utterance-level, we make use of heuristics to construct positive and negative samples for contrastive learning. The assumption is that one speaker mostly participates in only one session [1], and utterances in different conversations are naturally in different sessions. Suppose the speaker of u_i is $s(i)$ in the conversation U_i, we update the utterance level contrastive loss as below:

$$L'_u = \sum_{i=1}^{|\mathcal{U}|} \frac{-1}{|\mathcal{Y}'(i)|} \sum_{j \in \mathcal{Y}'(i)} log \frac{exp(\mathbf{v}_i \cdot \mathbf{v}_j/\tau'_1)}{\sum_{l \in \mathcal{N}'(i,j)} exp(\mathbf{v}_i \cdot \mathbf{v}_l/\tau'_1)},$$

where $\mathcal{Y}'(i) \equiv \{j \in U_i : s(j) = s(i), j \neq i\}$, and $\mathcal{N}'(i,j) \equiv \{l \in \mathcal{U}/U_i\} \cup \{j\}$. To sum up, the final objective for unsupervised training is as below:

$$L_{unsupervised} = L'_u + \eta L'_s, \qquad (2)$$

where η is a hyper-parameter to adjust the contribution of different factors.

After the representation learning, we may use various methods to decide the session number k for each conversation, such as the Elbow algorithm (Thorndike, 1953), or Silhouette algorithm (Rousseeuw, 1987). Empirically, we find the Elbow algorithm works slightly better. Based on the predicted K, we simply run the K-Means clustering to obtain the session assignments.

4 Experiments

4.1 Dataset

We train and evaluate our models on two large-scale annotated datasets. The first dataset is the Ubuntu IRC dataset (Kummerfeld et al., 2019), which consists of 153/10/10 intermingled dialogues in the train/validation/test set. Each dialogue is extracted from the Ubuntu IRC technical support channel and has a length of 250 or 500. Following (Liu et al., 2020), we cut each dialogue into dialogue segments of length 50, reorder the ground truth session labels, and get 1,737/134/104 dialogues in the train/validation/test split. The maximum session number is 14 for the Ubuntu IRC dataset. The second dataset is the Movie Dialogue dataset (Liu et al., 2020). The dialogues are generated by extracting sessions from 869 movie scripts and manually intermingling the sessions. There are 29,669/2,036/2,010 dialogues train/validation/test split. The maximum session number is 6.

4.2 Training Details

We initialize the word embeddings with 300-dimensional Glove vectors (Pennington et al., 2014) and set the hidden state size of BiLSTM to be 300. The utterance embedding size after the co-attention layer will also be 300. The maximum length of an utterance after tokenization is set to 50. In supervised training, the hyperparameter α and β that controls the weights are configured as 0.4 empirically, while γ is set to 0.2. We adopt a batch size of 16 and use Adam Optimizer (Kingma and Ba, 2015) with an initial learning rate of 5e-5. We run ten epochs until convergence. In unsupervised training, the hyper parameters will be the same and η is set to 0.4. While certain hyper-parameters such as Glove embedding size are set according to the default practice of previous works, other hyper-parameters such as batch size and maximum sequence length are determined empirically. In particular, the weight parameters α, β, γ, and η are tuned with grid search.

[1] Only 20% of speakers will join multiple sessions on the Ubuntu IRC dataset.

		Ubuntu IRC			Movie Dialogue		
		NMI	ARI	$Shen-F$	NMI	ARI	$Shen-F$
Supervised	*Weighted SP**	0.253	0.026	0.333	0.184	0.041	0.523
	*CISIR**	0.466	0.034	0.408	0.205	0.065	0.538
	*BERT**	0.546	0.082	0.439	0.256	0.110	0.569
	Transition	0.611	0.198	0.538	0.329	0.248	0.650
	DialBERT	0.675	0.245	0.605	0.362	0.180	0.597
	+ cov	**0.696**	0.275	0.615	0.328	0.180	0.608
	+ feature	0.671	0.216	0.586	-	-	-
	+ future context	0.671	0.226	0.591	0.358	0.174	0.587
	StructBERT	0.678	0.371	0.675	0.446	0.327	0.695
	w/max-pooling	0.677	**0.379**	0.681	0.448	0.327	0.695
	Bi-CL (Ours)	0.624	0.360	**0.707**	**0.575**	**0.382**	**0.747**
Unsupervised	*Co-Training*	0.540	0.182	0.456	0.290	0.217	0.592
	- pseudo data	0.531	0.168	0.409	0.279	0.201	0.576
	Zeroshot	0.597	0.212	0.578	-	-	-
	+ augment data	**0.639**	0.292	0.642	-	-	-
	Bi-CL (Ours)	0.607	**0.345**	**0.704**	**0.581**	**0.366**	0.689
	w/Silhouette	0.606	0.343	0.704	0.562	0.352	**0.698**

Table 1: Results on the Ubuntu IRC Dataset and the Movie Dialogue Dataset. * indicates that the statistics are taken from (Liu et al., 2020). Note the results of DialBERT + feature on the Movie Dialogue Dataset is not available since the dataset does not provide the corresponding features.

4.3 Metrics

We adopted three popular metrics to evaluate the disentanglement result: Normalized Mutual Information (*NMI*), Adjusted Random Index (*ARI*) (Hubert and Arabie, 1985), and Shen-F score (*Shen-F*) (Shen et al., 2006). Both *NMI* and *ARI* measures the similarity between the ground truth clusters and the predicted clusters for each conversation and a higher value indicates higher degree of matching. The difference is that *ARI* is based on counting pairwise links between utterances that exist in both ground truth and predictions, while *NMI* is more about the cluster level since it uses entropy conditioned on clusters. *Shen*-F is a F-1 score to measure how well utterances in the same ground truth cluster are grouped in the predicted clusters, and a higher value indicates higher cluster quality.

4.4 Baseline Models

We evaluate on both supervised and unsupervised settings. The baselines include both the traditional two-stage based and end-to-end approaches.

Supervised Baselines: The majority of methods need supervision. *Weighted SP* (Shen et al., 2006) adopts a single pass greedy decoding to add and cluster utterances sequentially based on normalized TF-IDF vectors. *CISIR* (Jiang et al., 2018) uses Hierarchical CNN to encode utterances and compute score of pairs. *Transition* (Liu et al., 2020) is an end-to-end online approach where each utterance is encoded and compared with the existing session states to determine assignments. *DialBERT* (Li et al., 2020b) gains from hierarchical Pre-Trained model for better performance. *StructBERT* (Ma et al., 2022) emphasizes structural characteristics in modeling and is the current state-of-the-art.

Unsupervised Baselines: When no labeled data is available, *Co-Training* (Liu et al., 2021) leverages a message-pair classifier and session classifier to build up a co-training scheme. *Zeroshot* (Chi and Rudnicky, 2021) learns from a closely related response selection task.

4.5 Main Results

We report the main results for all compared methods in Table 1. Generally speaking, the proposed *Bi-CL* method performs better than all the other baselines on both the Ubuntu IRC and Movie Dialogue datasets in most evaluation metrics. Note that some of these baselines are based on large-scale pre-trained language model BERT which has shown superior performance on various NLP tasks, our model is only based on the relatively lightweight bidirectional LSTM model. This situation, in some sense, signals the effectiveness of our bi-level contrastive learning design.

More specifically, under the supervised setting, the proposed *Bi-CL* method constantly outperforms other methods on the Movie Dialogue dataset

		Ubuntu IRC			Movie Dialogue		
		NMI	ARI	$Shen-F$	NMI	ARI	$Shen-F$
Supervised	*Bi-CL*	0.624	0.360	0.707	0.575	0.382	0.747
	w/gold K	0.611	0.379	0.716	0.614	0.421	0.763
	- L_u	0.548	0.266	0.656	0.508	0.335	0.736
	- L_s	0.566	0.323	0.684	0.541	0.340	0.731
	- L_m	0.596	0.345	0.697	0.542	0.341	0.731
	- L_k	0.612	0.282	0.643	0.133	0.100	0.589
Unsup.	*Bi-CL*	0.607	0.345	0.704	0.581	0.366	0.689
	w/gold K	0.608	0.374	0.714	0.609	0.420	0.763
	- L'_u	0.516	0.158	0.571	0.360	0.161	0.624
	- L'_s	0.607	0.337	0.640	0.570	0.354	0.683

Table 2: Ablation study on different design components of the proposed *Bi-CL* method under both settings.

across all metrics. It also performs the best on the Ubuntu IRC dataset regarding the metric *Shen-F*. This demonstrates the effectiveness of our Bi-level contrastive learning design for conversation disentanglement. We notice that *DialBERT* and *StructBERT* obtain better *NMI* results on Ubuntu IRC than our method. This is because these methods have special designs to model pairwise relations in a more fine-grained manner, by utilizing additional dialogue features such as the time of each utterance in Ubuntu IRC. Our model omits such data-specific features for model generalizability. In *StructBERT*, the ground truth reference dependencies are leveraged for structural characterization, hence we observe the best *ARI* performance. However, our model indeed surpasses the others on *Shen-F*. Although the margin between the result of our model (0.707) and that of *StructBERT w/max-pooling* (0.681) is smaller than the relatively large margin between the results of *Transition*, *DialBERT*, and *StructBERT*, our model's gain is shown to be statistically significant. We conduct a significance test by running our model in the same setting for 10 times and obtain standard deviation of *NMI* (0.00426), *ARI* (0.00286), and *Shen-F* (0.00186). With the significance level of 0.05, our result for *Shen-F* is significantly superior to the most competitive baseline.

Under the unsupervised setting, our model again excels except for *NMI* in Ubuntu IRC. This might be because the model *Zeroshot* has access to more augmented data from the same data source. However, it still performs worse than *Bi-CL* in *ARI* and *Shen-F*. Note that our model outperforms the baselines with a significant margin on the Movie Dialogue dataset. Again, this implies our model's generalizability. The model *Zeroshot* does not have results on the Movie Dialogue dataset. It relies on same source data to train response selection model, but such data is not available. We also put the our model's results with Silhouette algorithm as the K predictor. There is a slight drop in performance, which can be attributed to the lower prediction accuracy presented on Table 3.

A common pattern shared across the above settings is that while the baselines' results are typically much higher in Ubuntu IRC than in Movie Dialogue, *Bi-CL* performs stably across the two datasets. This is consistent with our previous observation that *Bi-CL* is independent of many features in Ubuntu IRC that are heavily utilized but often not available for other data sources. Moreover, the performance gap between the supervised and unsupervised versions of *Bi-CL* is relatively small, suggesting that it also relies less on labels. These demonstrate the potential of the model to be applied widely.

4.6 More Analysis

We further carry out ablation studies on various design components and provide more analysis on the prediction of session number K.

4.6.1 Ablation Study

We conduct ablation studies to investigate how each model component affects its effectiveness. As shown in Table 2, we observe that in the supervised setting, removing L_k leads to the most significant performance drop, with the gaps of 0.442, 0.282 and 0.158 in *NMI*, *ARI* and *Sehn-F* on the Movie Dialogue. This is because it makes predicting *K* degenerate into a random guess. Also, we observe that L_u has the second most impact. For example, it reaches the lowest performance on Ubuntu IRC regarding *NMI* and *ARI*. Removing the other components has a smaller impact and the model can still generate reasonable result.

In unsupervised setting, removing L'_u undermines the model significantly on *ARI* since it removes pairwise contrastive learning on the utterance-level that helps to model local relations. Removing L'_s tends to have a milder impact, but it still undermines the results to a certain extent. The above results imply that the utterance-level loss captures local pairwise relations well and the session-level loss also has positive contribution to learning cluster-friendly utterance representations.

	Ubuntu IRC		Movie Dialogue	
	ACC	MAE	ACC	MAE
Supervised	0.272	1.389	0.682	0.330
Silhouette	0.166	2.085	0.251	1.074
Elbow	0.203	1.731	0.227	1.195

Table 3: Accuracy (*ACC*) and Mean Absolute Error (*MAE*) of the predictions given by the *K* predictors.

4.6.2 Prediction of *K*

Predicting the session number *K* is crucial for our model since it directly affects the clustering results. We hence replace the predicted *K* with the ground truth *K* in training and inference, resulting in a moderate performance boost (*w/gold K*) in both settings as shown in Table 3. We also observe that the performance gaps between model using predicted *K* and ground truth *K*. This show that the model with the predicted *K* can still generate relatively satisfactory results and the performance of *K* prediction is relatively good. We show the *ACC* and *MAE* of predicted *K* in Table 3. It indicates that the supervised predictor works better which is reasonable, and the unsupervised methods such as Silhoutte and Elbow perform similarly. This might be because both of them only work on utterance features. Introducing other side information from conversation might further boost the performance.

Another observation is that *NMI* on Ubuntu IRC has a decrease when gold *K* value is adopted in supervised setting. While it is counter-intuitive, it may actually be caused by large number of sessions that contains only one utterance in this dataset.

5 Conclusion

We studied disentanglement on multi-party conversations and proposed a general model that works in both supervised and unsupervised learning settings. It is trained with a Bi-Level contrastive learning mechanism to bring utterances in the same session closer and encourage utterances to flock around their session centers. At the same time, we aim to pull utterances from different sessions further apart by contrasting each utterance with negative samples. The obtained representations naturally fit to the clustering scheme for session predictions. Consequently, K means is used during inference to predict the sessions. Our model is evaluated on the largest benchmark dataset Ubuntu IRC and the latest benchmark dataset Movie Dialogue. Experimental results show new SOTA performance results and advancements compared to previous works. Additionally, the stability of our model across different datasets, as well as different training schemes with or without session labels, shows its potential to be applied in a general setting,

6 Limitations

Our work has the following limitations. Firstly, although bidirectional LSTM is more light-weight and obtains reasonable performance for our task, an easy extension is to explore how pre-trained language models such as BERT would further affect the performance (Liao et al., 2021a). Secondly, the prediction of session number *K* is only based on conversation utterances. More advanced session number estimation model would be devised to capture more side information for more accurate *K* prediction. An alternative approach is to adopt different clustering algorithm such as CISIR (Jiang et al., 2018) that does not require the prediction of cluster number but instead has a universal, empirically determined threshold that controls the cluster size. Last but not least, our model has not been applied to dialogues of length longer than 50, and we have not verified its effectiveness of modeling longer dependency. This entails our future effort to adapt our model to a more general setting with longer conversation, more threads, and more complicated dialogue structures.

Acknowledgments

This research was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant.

References

Paul M. Aoki, Margaret H. Szymanski, Luke D. Plurkowski, James D. Thornton, and Allison Woodruff. 2006. Where's the "party" in "multi-party"?: analyzing the structure of small-group sociable talk. In *CSCW*, pages 393–402.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 33:9912–9924.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607.

Ta-Chung Chi and Alexander Rudnicky. 2021. Zero-shot dialogue disentanglement by self-supervised entangled response selection. In *EMNLP*, pages 4897–4902.

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

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

Hao Fei, Jingye Li, Shengqiong Wu, Chenliang Li, Donghong Ji, and Fei Li. 2022a. Global inference with explicit syntactic and discourse structures for dialogue-level relation extraction. In *IJCAI*, pages 4107–4113.

Hao Fei, Shengqiong Wu, Meishan Zhang, Yafeng Ren, and Donghong Ji. 2022b. Conversational semantic role labeling with predicate-oriented latent graph. In *IJCAI*, pages 4114–4120.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*, pages 6894–6910.

Jia-Chen Gu, Tianda Li, Quan Liu, Xiaodan Zhu, Zhen-Hua Ling, and Yu-Ping Ruan. 2020. Pre-trained and attention-based neural networks for building noetic task-oriented dialogue systems. In *AAAI Workshop on DSTC8*, pages 1–9.

Gaoyang Guo, Chaokun Wang, Jun Chen, and Pengcheng Ge. 2018. Who is answering to whom? finding "reply-to" relations in group chats with long short-term memory networks. In *EDB*, pages 161–171.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pages 1735–1742.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738.

Olivier Henaff. 2020. Data-efficient image recognition with contrastive predictive coding. In *ICML*, pages 4182–4192.

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

Shenhao Jiang, Animesh Prasad, Min-Yen Kan, and Kazunari Sugiyama. 2018. Identifying emergent research trends by key authors and phrases. In *COLING*, pages 259–269.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *NeurIPS*, 33:18661–18673.

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

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

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

Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. 2020a. Prototypical contrastive learning of unsupervised representations. In *ICLR*.

Tianda Li, Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. 2022. Conversation-and tree-structure losses for dialogue disentanglement. In *DialDoc-ACL*, pages 54–64.

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 arXiv:2004.03760*.

Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. 2021. Contrastive clustering. In *AAAI*, pages 8547–8555.

Lizi Liao, Le Hong Long, Yunshan Ma, Wenqiang Lei, and Tat-Seng Chua. 2021a. Dialogue state tracking with incremental reasoning. *TACL*, 9:557–569.

Lizi Liao, Le Hong Long, Zheng Zhang, Minlie Huang, and Tat-Seng Chua. 2021b. Mmconv: an environment for multimodal conversational search across multiple domains. In *SIGIR*, pages 675–684.

Lizi Liao, Yunshan Ma, Xiangnan He, Richang Hong, and Tat-seng Chua. 2018. Knowledge-aware multimodal dialogue systems. In *ACM MM*, pages 801–809.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *ICLR*.

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

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

Ryan Thomas 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).

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2022. Structural characterization for dialogue disentanglement. In *ACL*, pages 285–297.

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

Ishan Misra and Laurens van der Maaten. 2020. Self-supervised learning of pretext-invariant representations. In *CVPR*, pages 6707–6717.

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

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Iulian Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, volume 31.

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

Yuming Shen, Ziyi Shen, Menghan Wang, Jie Qin, Philip Torr, and Ling Shao. 2021. You never cluster alone. *NeurIPS*, 34:27734–27746.

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 *EMNLP-IJCNLP*, pages 6456–6461.

Robert L Thorndike. 1953. Who belongs in the family. In *Psychometrika*.

Lidan Wang and Douglas W Oard. 2009. Context-based message expansion for disentanglement of interleaved text conversations. In *NAACL*, pages 200–208.

Weishi Wang, Steven CH Hoi, and Shafiq Joty. 2020. Response selection for multi-party conversations with dynamic topic tracking. In *EMNLP*, pages 6581–6591.

Yuxia Wu, Lizi Liao, Gangyi Zhang, Wenqiang Lei, Guoshuai Zhao, Xueming Qian, and Tat-Seng Chua. 2022. State graph reasoning for multimodal conversational recommendation. *TMM*.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. In *ACL*, pages 5065–5075.

Chenchen Ye, Lizi Liao, Fuli Feng, Wei Ji, and Tat-Seng Chua. 2022a. Structured and natural responses co-generation for conversational search. In *SIGIR*, pages 155–164.

Chenchen Ye, Lizi Liao, Suyu Liu, and Tat-Seng Chua. 2022b. Reflecting on experiences for response generation. In *ACM MM*, pages 5265–5273.

Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. 2019. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, pages 6210–6219.

Tao Yu and Shafiq Joty. 2020. Online conversation disentanglement with pointer networks. pages 6321–6330.

Zheng Zhang, Lizi Liao, Minlie Huang, Xiaoyan Zhu, and Tat-Seng Chua. 2019. Neural multimodal belief tracker with adaptive attention for dialogue systems. In *WWW*, pages 2401–2412.

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 *AAAI*, pages 9741–9748.

Rongxin Zhu, Jey Han Lau, and Jianzhong Qi. 2021. Findings on conversation disentanglement. pages 1–11.

A Appendix

Prototypical Contrastive Learning was originally introduced in (Li et al., 2020a) to learn image representations. An Expectation-Maximization framework is constructed, where the E step estimates the distribution of the prototypes via K-Means clustering and the M step maximizes the likelihood of the network parameters. Similarly, consider a dialogue with n utterances $U = \langle u_1, ..., u_n \rangle$ that are embedded as $\langle \mathbf{v}_1, ..., \mathbf{v}_n \rangle$. Denote the embedding network parameters as θ, the objective is to find the optimal parameters θ^* that maximizes the log likelihood of the utterance representations:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{|U|} \log p(\mathbf{v}_i|\theta).$$

Define the set of prototypes in the dialogue as $C = \{c_j\}_{j=1}^m$, which are the centroids of the clusters generated by the K-Means algorithm applied on the utterance embeddings. The likelihood for utterance u_i can be written as the summation of the joint probability for u_i being observed and belong to each prototype c_j. Hence, we have:

$$\begin{aligned}\theta^* &= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{|U|} \log \sum_{\mathbf{c}_j \in C} p(\mathbf{v}_i, \mathbf{c}_j|\theta) \\ &= \sum_{i=1}^{|U|} \log \sum_{\mathbf{c}_j \in C} Q(\mathbf{c}_j) \frac{p(\mathbf{v}_i, \mathbf{c}_j|\theta)}{Q(\mathbf{c}_j)} \\ &\geq \sum_{i=1}^{|U|} \sum_{\mathbf{c}_j \in C} Q(\mathbf{c}_j) \log \frac{p(\mathbf{v}_i, \mathbf{c}_j|\theta)}{Q(\mathbf{c}_j)},\end{aligned}$$

where Jensen's inequality is applied and we have $Q(\mathbf{c}_j) = p(\mathbf{c}_j|\mathbf{v}_i, \theta)$. By ignoring the constant $-\sum_{i=1}^n \sum_{\mathbf{c}_j \in C} Q(\mathbf{c}_j) \log Q(\mathbf{c}_j)$, the transformed objective becomes to maximize:

$$L = \sum_{i=1}^{|U|} \sum_{\mathbf{c}_j \in C} Q(\mathbf{c}_j) \log p(\mathbf{v}_i, \mathbf{c}_j|\theta). \qquad \text{(A.1)}$$

A.1 E step

In this step, we estimate $Q(\mathbf{c}_j) = p(\mathbf{c}_j|\mathbf{v}_i, \theta)$, which is the likelihood for u_i to be allocated to the cluster with c_j as the centroid. We model $p(\mathbf{c}_j|\mathbf{v}_i, \theta)$ as $\mathbb{1}(u_i \in c_j)$, which is 1 if u_i is allocated to the cluster corresponding to c_j by the K-means algorithm, and 0 otherwise.

A.2 M step

In this step, we model Equation A.1 and derive the maximization objective. Note that:

$$\begin{aligned}p(\mathbf{v}_i, \mathbf{c}_j|\theta) &= p(\mathbf{v}_i|\mathbf{c}_j, \theta)p(\mathbf{c}_j|\theta) \\ &= \frac{1}{m} \cdot p(\mathbf{v}_i|\mathbf{c}_j, \theta),\end{aligned}$$

since we assume any unseen utterance has an equal probability to belong to any session ($p(\mathbf{c}_j|\theta) = \frac{1}{m}$).

Additionally, we assume that the distribution of an utterance u_i around each prototype c_j is an isotropic Gaussian distribution. Therefore, we have:

$$p(\mathbf{v}_i|\mathbf{c}_j, \theta) = \frac{\exp -\frac{(\mathbf{v}_i - \mathbf{c}_j)^2}{2\sigma_j^2}}{\sum_{l=1}^m \exp -\frac{(\mathbf{v}_i - \mathbf{c}_l)^2}{2\sigma_l^2}}.$$

We apply l_2 normalization to $\mathbf{v}_i$ and $\mathbf{c}_l$, so that $(\mathbf{v}_i - \mathbf{c}_l)^2 = 2 - 2\mathbf{v}_i \cdot \mathbf{c}_l$. As a result, we have:

$$\begin{aligned}&\sum_{i=1}^{|U|} \sum_{\mathbf{c}_j \in C} Q(\mathbf{c}_j) \log p(\mathbf{v}_i, \mathbf{c}_j|\theta) \\ &= \sum_{i=1}^{|U|} \sum_{c_j \in C} \mathbb{1}(u_i \in c_j) \log \frac{1}{m} p(\mathbf{v}_i|\mathbf{c}_j, \theta) \\ &= \sum_{i=1}^{|U|} \log \frac{1}{m} \frac{\exp(\mathbf{v}_i \cdot \mathbf{c}_i/\phi_i)}{\sum_{l=1}^m \exp(\mathbf{v}_i \cdot \mathbf{c}_l/\phi_l)},\end{aligned}$$

where ϕ_l indicates the concentration level of the utterance embedding around c_l. Here, ϕ_l is set as a constant across different prototypes since there are limited number of utterances in the dialogue.

To further enable the learning of contrastive features on different granularity, we cluster the utterance embeddings M times with cluster number ranging from 1 to M and update the network parameters with prototypes that encodes hierarchical structure. Consequently, we can write the optimal parameter as:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} -\frac{1}{M} \sum_{i=1}^{|U|} \sum_{m=1}^{M} \frac{\exp(\mathbf{v}_i \cdot \mathbf{c}_i/\tau_2')}{\sum_{l=1}^m \exp(\mathbf{v}_i \cdot \mathbf{c}_l/\tau_2')},$$

where τ_2' is a small constant.