Unsupervised Learning of Hierarchical Conversation Structure

Bo-Ru Lu[♠] **Yushi Hu**[♠] **Hao Cheng**[♡] **Noah A. Smith**[♠♢] **Mari Ostendorf**[♠]
[♠]University of Washington [♡]Microsoft Research [♢]Allen Institute for AI
{roylu,yushihu,ostendor}@washington.edu
chehao@microsoft.com nasmith@cs.washington.edu

Abstract

Human conversations can evolve in many different ways, creating challenges for automatic understanding and summarization. Goal-oriented conversations often have meaningful sub-dialogue structure, but it can be highly domain-dependent. This work introduces an unsupervised approach to learning hierarchical conversation structure, including turn and sub-dialogue segment labels, corresponding roughly to dialogue acts and sub-tasks, respectively. The decoded structure is shown to be useful in enhancing neural models of language for three conversation-level understanding tasks. Further, the learned finite-state sub-dialogue network is made interpretable through automatic summarization.

1 Introduction

Increasingly, language understanding applications involve conversational speech and text. Much attention has recently been directed at human-agent dialogue systems, including virtual assistants, interactive problem solving, and information seeking tasks (e.g., conversational question answering). However, automatic understanding of human-human conversations is also of interest for problems such as call-center analytics, conversation outcome prediction, meeting summarization, and human-agent interaction involving multiple people. The focus of this paper is on human-human conversation understanding.

Like written documents, goal-oriented conversations tend to have structure (openings, context setting, problem solving, etc.). However, in human-human conversations (both text and speech), participant roles factor into the structure, and the structure is less rigid due to the need to accommodate miscommunications and varying objectives. Yet, most work on conversational systems treats dialogues like written text, i.e., the dialogue history is a linear sequence of text. In this paper, we explore unsupervised learning strategies for adding structural information to a state-of-the-art hierarchical transformer-based model of text.

Linguistic analysis of conversations often involves associating speaker utterances with dialogue acts (DAs), e.g., question, statement, backchannel, clarification, etc. (Jurafsky et al., 1997; Core and Allen, 1997), and segmenting the conversation into nested subsequences of participant turns that reflect a common topic or conversational goal (Grosz and Sidner, 1986). Past studies have explored using such structure, particularly DAs, to improve automated human-agent dialogues. Here, we use hierarchical structure (both turn-level DA labels and sub-dialogue states) to improve classification of human-human conversations. Specifically, we introduce **T**hree-stream **Hie**rarchical **Tra**nsformer (THETA), which integrates transformer representations of the DA and sub-dialogue state sequences into a hierarchical transformer (HT) (Santra et al., 2021; Pappagari et al., 2019) operating on the original text. In addition to improving performance, the use of discrete structural cues in classification can support conversation analysis. For example, we can identify seller strategies that are more likely to lead to a successful outcome or use the sub-dialogue state sequence to summarize frequently visited states in unsuccessful interactions.

Since hand-annotation of structure can be costly and inventories vary across tasks, there is substantial interest in unsupervised learning of structure for specific task domains. Here, the approach to structure learning involves two steps. First, we use a clustering algorithm to learn a mapping of utterance embeddings to discrete categories, which serve as an unsupervised version of DAs. Each conversation is then represented by the discrete sequence of cluster identifiers (IDs) associated with the sequence of utterances. Using the collection of discretized conversations, we automatically learn

We release our code for experiments at https://github.com/boru-roylu/THETA.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5657–5670

the topology of a latent finite-state model over these sequences, i.e., a hidden Markov model (HMM), using a greedy state-splitting algorithm that maximizes the likelihood of the sequence data without requiring any annotations. The states of the HMM correspond to different sub-dialogues that may be associated with specific topics, strategies or subtasks. The sub-dialogue structure of a new conversation is identified by finding the most likely state sequence given that discretized utterance sequence.

The learned structure is assessed in experiments on three conversation-level classification tasks: buyer/seller negotiation outcomes on CRAIGSLISTBARGAIN (He et al., 2018), conversation category in the Action-based Conversations Dataset (ABCD) (Chen et al., 2021), and client callback prediction in a private call center corpus. In each task, we find that a combination of both utterance-level category and sub-dialogue state information lead to improved performance. Further, we use automatically generated descriptions of the clusters and sub-dialogue states to provide an interpretable view of the finite-state topology and a summarized view of a conversation. Anecdotally, we find that this structure lends insights into how participant strategies (state paths) are associated with different conversation outcomes.

The contributions of this work are as follows. First, we introduce a simple unsupervised approach to learn a hierarchical representation of conversation structure that includes turn-level labels and sub-dialogue segmentation, accounting for participant role. Using three conversation-level classification tasks, we demonstrate that integrating the structural information into a state-of-the-art hierarchical transformer consistently improves performance. Lastly, we show how the discrete representation of structure combined with automatic summarization can provide a mechanism for interpreting what the model is learning or for conversation summarization and analytics.

2 Method

As shown in Figure 1, THETA represents the sequence of turns in a conversation using: i) a hierarchical transformer (HT) operating on a turn-segmented word sequence, ii) a transformer operating on a sequence of turn-level DAs, and iii) a separate transformer operating on a sequence of sub-dialogue states derived from the DAs. The conversation-level vectors produced by the three transformers are concatenated and used in a final task-specific layer for conversation classification tasks. The HT alone is the state-of-the-art model for conversation-level tasks. The DA and sub-dialogue states comprise the structural information that enhances the HT for improving performance of the end task. In addition, the discrete nature of the structure representation provides a mechanism for analyzing the conversation classes via summarization of utterances associated with the DA labels or sub-dialogue states.

2.1 Model Components

Definitions More formally, each dialogue consists of a sequence of words (or tokens) $X = [x_1, \ldots, x_T]$ associated with T customer/agent (or seller/buyer) utterances, where x_t is the subsequence of words associated with the tth utterance.[1] The word sequence is decorated with three special tokens: `[CLS]`, `[PTY]` and `[SEP]`, where `[PTY]` indicates the utterance speaker role (`[AGT]` for agent/seller and `[USR]` for customer/buyer). The word sequence X is mapped to two sequences of utterance-level embeddings $U^v = [u_1^v, \ldots, u_T^v]$, $v \in \{\text{HT}, \text{DA}\}$. The vector u_t^{HT} is output from the last layer of the HT that is used to derive the text-based conversation-level vector $\mathbf{U}$. The vector u_t^{DA} is the output of a separate transformer, which is then mapped to a DA category c_t to produce the sequence $C = [c_1, \ldots, c_T]$. The sequence C is associated with a hidden subdialogue sequence that is represented using the HMM state sequence $S = [s_1, \ldots, s_T]$. Additional transformers derive conversation-level vectors $\mathbf{C}$ and $\mathbf{S}$ from C and S, respectively. THETA enhances the conversation representation by concatenating $\mathbf{U}$, $\mathbf{C}$ and $\mathbf{S}$ together for input to a task-specific layer.

Hierarchical Transformer The hierarchical transformer (Pappagari et al., 2019) has been shown to be useful for classifying long documents (like customer support conversations), which exceed the length limits placed on transformer-based models due to the quadratic complexity of the self-attention module. At a high level, two transformer blocks, a lower utterance transformer and an upper conversation transformer are stacked together for encoding dialogues. Here, the utterance-level transformer first encodes utterances into utterance embeddings, one for each utterance. In

[1] We use the term "utterance" although some conversations involve text-based interactions.

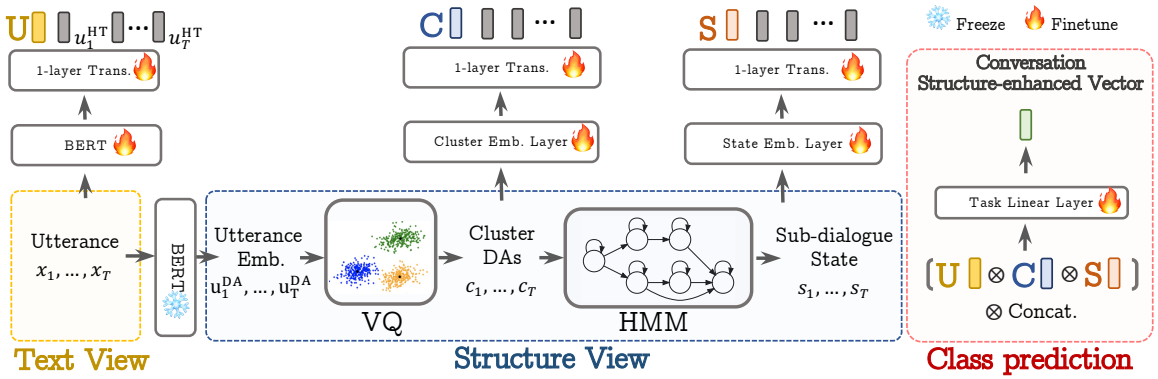


Figure 1: Overview of THETA conversation encoding. The text of each utterance text is encoded by BERT, and a 1-layer transformer further contextualizes utterance embeddings to generate the text vector **U**. For structure, utterances are mapped to K-means dialogue acts (DAs), which are input to an HMM to decode sub-dialogue states. 1-layer transformers are applied to sequences of DAs and sub-dialogue states, yielding cluster vector **C** and state vector **S**. The concatenation of **U**, **C** and **S** is fed into a linear layer to obtain the structure-enhanced vector for the predictive task. For simplicity, Emb. and Trans. stand for embedding and transformer, respectively.

this case, the first contextualized token embedding as the utterance embedding, which corresponds to the sentence-level `[CLS]` token. The sequence of utterance embeddings augmented with a conversation-level `[CLS]` token are then fed as inputs to another one-layer conversation-level transformer to further contextualize the vector sequence. We use the output vector associated with the conversation-level `[CLS]` token as the conversation representation.

Dialogue Act Sequence Module To obtain the DA labels, we first derive an utterance embedding u_t^{DA} by mean pooling the final layer of the BERT transformer.[2] The resulting embedding is mapped to a DA class c_t using a vector quantization (VQ) approach: K-means clustering is used to learn the classes, and vectors are labeled at inference time by minimizing the Euclidean distance to cluster means. The number of clusters is treated as a hyperparameter of the overall model. We apply K-means clustering separately for utterances from the two different participant roles, so the DA index reflects the role. This simple approach is motivated by prior work on unsupervised learning of DA categories (Brychcin and Kral, 2017), which showed that K-means clustering gives a performance that is only slightly worse than HMM-based learning.

In linguistic analyses, a turn can contain a sequence of DAs. Our work assigns a single DA to a user turn, as in other work using unsupervised learning as well as the negotiation data set that we report results on. Since the prior work often uses "dialogue act" for turn-level labels, we have chosen to use the DA term here, acknowledging the abuse of terminology. For complex tasks like the call center data (and other data with real users), the turns will involve multiple dialogue acts, in which case a large number of clusters is useful.

Sub-Dialogue Sequence Module The DA sequence C is input to a hidden Markov model (HMM) to derive the sub-dialogue structure. An HMM is a statistical model that characterizes an observation sequence C in terms of a discrete, latent (hidden) Markov state sequence S,

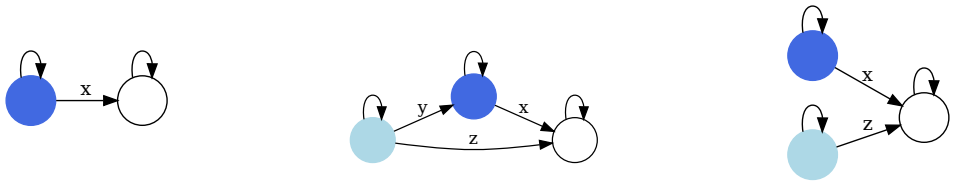
$$P(C) = \sum_{\text{all } S} P(C, S)$$
$$= \sum_{\text{all } S} \pi(s_1) \prod_{t=1}^{T} \eta(c_t|s_t)\gamma(s_{t+1}|s_t),$$

where π, η, and γ are start-state, observation, and transition distributions, respectively. s_{T+1} is a dummy stopping state. The HMM is used to decode the hidden sub-dialogue state sequence S, which provides a segmentation of the conversation into different stages or sub-tasks in problem solving or negotiation. The HMM topology and parameters are derived using unsupervised learning as described in the next section.

2.2 Sub-Dialogue Structure Learning

Given a specified topology, inference and training algorithms for HMMs are well established (Murphy, 2012); the Viterbi algorithm gives the

[2] We also experimented with using the `[CLS]` token, but mean pooling gave better results.



(a) Before split. (b) Temporal split. (c) Contextual split.

Figure 2: The design of two split methods. The dark-blue state is chosen to be split. The light-blue state is the new state after split. Transitions to other states are omitted for simplicity.

most likely state sequence, and the Expectation-Maximization (EM) algorithm is used for parameter estimation. To automatically learn the HMM topology, we apply a greedy state splitting algorithm (Ostendorf and Singer, 1997), which learns a left-to-right topology by constraining states to inherit the transition constraints of their parent. The standard objective is maximum likelihood of the DA sequence, which is unsupervised with respect to the conversation-level task.

Topology learning is outlined in Algorithm 1. The initial model has a 3-state left-to-right topology, initialized (assuming 70% of the conversation is associated with the middle state) and then iteratively trained until the improvement is lower than a fixed threshold or the iteration count exceeds some number. At each iteration, the state with the highest entropy of emission probability is chosen to be split. The topology can change into two new configurations corresponding to temporal and contextual splits (Figure 2). The EM algorithm is applied again on each configuration and the topology that leads to the higher likelihood is chosen. We iteratively conduct the splitting until the total number of states reaches the desired value (a hyperparameter). The HMMs of Ostendorf and Singer (1997) used continuous observation distributions. The splitting approach described below was designed for discrete distributions.

Algorithm 1 Topology Learning Algorithm

1: n: number of split. τ_i: topology after i split.
2: Initialization: Run the EM algorithm on 3-state initial topology τ_0.
3: **for** $i = 1$ **to** n **do**
4: **begin**
5: Select state $s \in \tau_{i-1}$ to split based on max entropy of observation distribution $\eta_{i-1}(c|s)$.
6: Apply temporal split and get new topology $\tau_{i,t}$.
7: Apply contextual split and get new topology $\tau_{i,c}$.
8: Run the EM algorithm on $\tau_{i,t}$ and $\tau_{i,c}$.
9: Select the topology with higher likelihood as τ_i.
10: **end**

Temporal split The temporal split provides more detailed sequential structure along a path. Figure 2(b) shows the result of a temporal split on the selected state (dark-blue) in Figure 2(a). The light-blue node is the new child state that inherits all incoming and outgoing edges and the transition probabilities of the dark-blue state except y and z. Edges y and z are initialized to $p_x/2$ where p_x is the probability of the original edge x of dark-blue state. The old incoming edges of dark-blue state are removed and outgoing edges are preserved.

Contextual split The contextual split allows for alternate sub-dialogue paths. Figure 2(c) illustrates the contextual split applied on the dark-blue state. The light-blue state inherits everything but the observation distribution of dark-blue state. With the aim of modeling different types of paths, when copying the observation probabilities to the light-blue state, we omit the top emission probability of the dark-blue node and set it to 0 and normalize the rest of probabilities. In terms of the transition probabilities, the light-blue state inherits all from the dark-blue one; $p_x = p_z$ where p_x and p_z are the transition probabilities of edges x and z.

2.3 Pre-Training and End-Task Training

Both for initializing the HT and for deriving the DAs, we use the transformer-based BERT model (Devlin et al., 2019) for encoding individual utterances u_t, pre-trained using masked language modeling and next-sentence prediction. Due to the style differences of dialogue data vs. written text, we apply domain-adaptive pretraining (DAPT) (Gururangan et al., 2020) to adapt BERT for dialogue applications. As shown later (section 3), adapting BERT with DAPT provides substantial improvement in terms of predictive power as well as optimization stability.

For the HT alone, supervised training involves learning the weights of the final task-level linear layer, the utterance-level transformer, and the word-

level transformer.

For THETA, supervised training involves learning the weights of the cluster- and state-level transformers, in addition to all updates associated with the HT component described above. The cluster sequences are obtained using the word-level transformer with DAPT and the associated cluster mapping obtained from unsupervised learning, i.e., without task-level finetuning. Similarly, there are no task-level supervision updates to the parameters associated with the HMM that is used to derive the state sequence.

3 Experiment

3.1 Datasets and Evaluation Metrics.

We use three datasets with conversation-level classification tasks to evaluate our model. The detailed statistics of the datasets are shown in Appendix B.

CRAIGSLISTBARGAIN (He et al., 2018) is a public negotiation dataset where buyers and sellers negotiate the prices of items on sale. In each conversation, the buyer has a target price in their mind and attempts to reach an agreement with the seller. Following previous work (Zhou et al., 2020; Joshi et al., 2021), we use the same list of 14 handcrafted utterance DAs and the 5-class sale-to-list price ratio labels provided in their code base. The 14 handcrafted utterance DAs are used as comparison to evaluate if our unsupervised version of DAs is learning good representations. Classification of sale-to-list price ratio is used as the downstream task, with accuracy as the evaluation criterion.

ABCD (Chen et al., 2021) is a public customer support dataset that is introduced to study customer service dialogues. In each conversation, an agent follows guidelines to help a customer solve their issue. Conversations are categorized with flows and subflows. Flows are broad categories, such as shipping issue, account access, or purchase dispute. Subflows comprise 96 fine-grained labels, for example, shipping status question, recover password, or invalid promotion code. Each conversation is annotated with a flow and a subflow. We use classification of the subflows as our conversation-level task. Macro and micro F1 scores are used to reflect the performance of imbalanced subflow classes.

CALL CENTER is a private collection of customer service conversations. Phone calls are automatically transcribed and private user information is anonymized. Conversations are annotated with a binary indicator as to whether or not there will be a callback within two days. (Such callbacks are an indicator that the problem was not solved in the call.) For the task of callback prediction, we measure area under the ROC curve (ROC AUC).

3.2 Implementation Details

Experimental Setup. We develop our K-means and HMMs using the packages Faiss (Johnson et al., 2019) and Pomegranate (Schreiber, 2018). The number of DAs and the size of the HMM state space are chosen separately for each dataset based on development set performance. We initialize and finetune our experiments based on uncased base model of BERT downloaded from HuggingFace (Wolf et al., 2020). We DAPT with dynamic whole-word masking (WWM) on 128-token segments for each dataset. During finetuning, the learning rate and warm-up steps are 1×10^{-5} and 0.1 epoch, respectively. Models are selected by the best score on the development set for each dataset. Further hyperparameter details are in Appendix A.

Model	% Acc.
FeHED	42.3
HED + RNN	47.9
HED + transformer	53.7
DIALOGRAPH	53.1
HT	54.1$\pm$ 2.4
THETA	**66.1**$\pm$ **1.0**

Table 1: Results on the test set of CRAIGSLISTBARGAIN in accuracy. For models studied in this paper (lower part), the median number is reported with standard deviation calculated based on 15 random runs.

	ABCD			CALL CENTER
	F1			
Model	Micro	Macro	Weighted	ROC AUC
HT	52.2	25.4	45.7	69.6
THETA	**62.8**	**39.1**	**59.9**	**71.3**

Table 2: Results on the test sets of ABCD and CALL CENTER datasets.

3.3 Comparison Systems

We use the hierarchical transformer (HT) as a baseline for all datasets in comparison to THETA. For CRAIGSLISTBARGAIN, we also include three additional baselines from two works (Zhou et al., 2020; Joshi et al., 2021) that employ the DAs extracted by heuristic methods; our systems use K-means to obtain primitive DAs.

	CraigslistBargain	ABCD			Call Center
		F1			
Model	Accuracy	Micro	Macro	Weighted	ROC AUC
HT w/o DAPT	48.0	15.4	4.2	9.4	68.4
HT	50.3	52.2	26.9	46.3	71.2
THETA (cluster only)	60.2	59.8	35.3	55.7	72.2
THETA (state only)	51.7	58.8	32.8	54.1	72.1
THETA	**61.3**	**62.6**	**38.6**	**59.5**	**72.8**

Table 3: Ablation on the development sets of CraigslistBargain, ABCD and Call Center datasets. All models with structure are statistically better than HT. THETA is better ($p < 0.01$) than the cluster-only alternative except for the Call Center.

FST-enhanced hierarchical encoder-decoder model (FeHED). FeHED (Zhou et al., 2020) uses an RNN-based sequence-to-sequence model with finite-state transducers for encoding sequences of strategies and DAs.

Hierarchical encoder-decoder (HED) + RNN or transformer. HED encodes dialogue utterances with a transformer (initialized from pretrained BERT), and the decoder generates the next response. An RNN or transformer encodes strategies and DAs. HED + RNN is based on the dialogue manager of He et al. (2018); Joshi et al. (2021) replace the RNN with a transformer.

DialoGraph. (Joshi et al., 2021). The state-of-the-art HED-based model on CraigslistBargain dataset leverages graph attention networks (GAT; Veličković et al., 2018) to encode strategies and DAs.

3.4 Prediction Results

Performance on Negotiation Dialogues. Table 1 reports the results of different systems on the test set of CraigslistBargain dataset. All models are based on the BERT-base model. HT with only text outperforms the state-of-the-art DialoGraph which leverages a graph-based representation of conversation structure. This verifies our hypothesis that DAPT with target data indeed improves BERT for dialogue tasks. Compared with HT, THETA achieves better prediction accuracy and smaller variance, which suggests that integrating the structure view helps stabilize training with different random seeds. THETA provides a 24.5% relative gain in accuracy over DialoGraph, setting a new state of the art. This further validates the advantage of our learned conversation structure for a predictive task.

Performance on Customer Support Domain. Similar to the results on the negotiation dialogue domain, Table 2 shows that conversation structure effectively enhances the performance in the customer service domain, ABCD and Call Center.

Ablation. Table 3 reports the results of ablating different components of THETA on the validation sets of all datasets. The first rows show that DAPT is useful on all tasks particularly for ABCD with its skewed class distribution. We also observe that THETA consistently achieves the best performance over all tasks. The cluster-based DA sequence provides more information than the sub-dialogue states, but incorporating all three views together leads to the best performance. Statistical significance is tested using bootstrap resampling (Efron and Tibshirani, 1993; Berg-Kirkpatrick et al., 2012).

Prior work (Zhou et al., 2020; Joshi et al., 2021) on CraigslistBargain use domain knowledge in rule-based annotation of DAs. To assess the use of K-means clusters for learning DAs, we also trained an HMM using the provided DAs. The resulting model obtained 66.5% accuracy on the test data, which is not significantly different the 66.1% results obtained using K-means (cf. Table 1).

4 Interpretation and Analysis

In this section, we leverage automatic summarization of clusters and states to derive insights into the learned conversation structure, both for interpretability of the model and for applications such as conversation analytics and summarization. As an example, we analyze fine-grained components from the learned topology, i.e., most frequent paths and individual state n-grams, to investigate their associations with different dialogue characteristics.

We apply graph-based unsupervised summarization (Boudin and Morin, 2013; Shang et al., 2018)

over utterances in each state (decoupling participant roles) and in each cluster. On CRAIGSLISTBARGAIN and ABCD, this leads to more than 3× reduction in conversation length.

Figure 3 shows the 8-state topology of CRAIGSLISTBARGAIN with selected state summaries. Based on the summaries, it is easy to see that `S1` and `S8` capture opening and closing DAs, respectively, while `S5` and `S6` correspond to different negotiation strategies. We also find that conversations with shorter paths are likely to involve a less experienced seller or lower buyer interest, e.g., 92% conversations with path `S1-S2-S8` lead to under listing sells. On the other hand, sellers that say offers are too low are more likely get better prices, e.g., 91% conversations with path `S1-S2-S3-S5-S7-S8`.

ABCD Table 4 shows an example with both cluster and state summaries. Based on the cluster summaries, we see that K-means learns typical DAs associated with customer service, e.g., information requests from the agent and the corresponding customer replies (`<name>`, `<email>`, `<address>`). States correspond to sub-dialogues where the agent follows certain protocols in resolving a sub-task (e.g., verifying account information). Alignment of flow labels with the most frequent paths through the HMM topology shows that paths are highly indicative of the corresponding dialogue flow. The high confusions are among certain flows, such as `storewide_query` and `single_item_query`, which one would expect to have have similar DAs.

5 Related Work

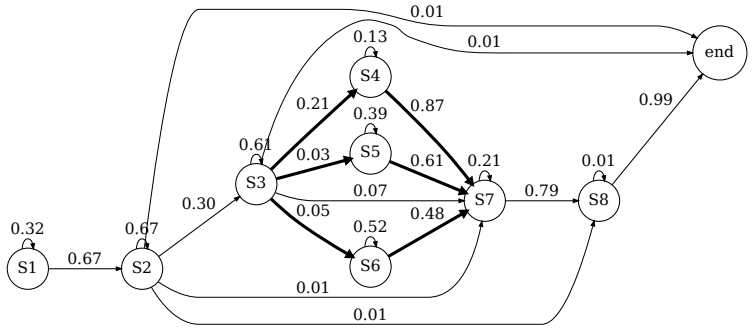
HMMs have been leveraged for learning structure in language for many years, such as in early work on inducing word-level part-of-speech tags (Merialdo, 1994). Accordingly, most work on unsupervised learning of both DAs and conversation structure leverages HMMs.

Unsupervised Learning of Dialogue Acts. Since dialogue act recognition can be thought of as a sentence-level tagging task, initial work on unsupervised learning of DAs was similar to word tagging, involving some use of language models or fully-connected HMMs to account for sequential dependency of labels. Ritter et al. (2010) use an HMM with a factored state space with a topic model to decouple speech act from topic characteristics. The observation model η in the HMM is a bag of words (unigram) model. The approach was later extended by incorporating speaker information (Joty et al., 2011; Paul, 2012). Brychcin and Kral (2017) further extend this work with a Gaussian mixture observation model (GMM) where the utterance representation is the average of GloVe word embeddings. They compare the results to a simple K-means clustering, which is not as effective as the HMM but gives similar results to the method proposed by Ritter et al. (2010) when applied to the Switchboard corpus. Hierarchical clustering of delexicalized utterance embeddings is used by Gunasekara et al. (2019), who use domain knowledge in preprocessing to identify phrases such as "Indian food" as "CUISINE_TYPE," for example. Our work on utterance categorization is similar to the K-means approach in Brychcin and Kral (2017), but we use more recent transformer-based utterance embeddings.

Unsupervised Learning of Dialogue Structure. Task- or goal-oriented conversations typically have structure above the level of the sentence in that a sequence of turns are associated with a common function. In more complex conversations, the structure can be hierarchical, with tasks and sub-tasks. Bangalore et al. (2008) used a parsing model to automatically recognize dialogue acts and segment a conversation into sub-tasks, leveraging hand-annotations of both DAs and sub-tasks. Since sub-task structure varies depending on the task and there is little hand-annotated data, most work has focused on unsupervised approaches with a flat segmentation. Note that the problem of unsupervised learning here involves jointly recognizing sub-dialogue segment boundaries, learning an inventory of sub-dialogue types, and learning (or constraining) the sequential structure of these types.

Early work on unsupervised learning used fully-connected HMMs to identify structure in documents (Barzilay and Lee, 2004) for extractive summarization and information ordering. The observation model was based on word bigrams with the aim of capturing topic coherent segment. A similar idea is applied to task-oriented dialogues using latent Dirichlet allocation for the observation model (Zhai and Williams, 2014).

Studies that leverage constrained left-to-right HMM technologies include (Althoff et al., 2016), which aimed to learn stages/strategies of counselors in mental health counseling, and (Ramanath et al., 2014), which used a hidden semi-Markov



S1	buyer: hi i am interested seller: hi how are you interested ?
S2	buyer: how about its condition ? seller: it s in great condition .
S3	buyer: i can do <price>. i can pick it up seller: that s too low. i can go with <price>
S5	buyer: i can do <price>. <offer> seller: the lowest i can do is <price>
S6	buyer: that s a deal. i can do that seller: great. that s a deal
S8	buyer: <accept> seller: <accept>

Figure 3: The 8-state topology on CRAIGSLISTBARGAIN dataset. The thicker edges indicate higher levels of negotiation success; in contrast, the thinner edges represent lower levels. Due to space limitations, only 6 state summaries are shown. The detailed topology with all cluster and state summaries is in Appendix D.

Party	Utterance	Cluster Summary	State Summary
Agent	Welcome to AcmeBrands! How can I help you?	How can I help you?	A: How can I help you today?
Customer	Hello, I would like to change my shipping details as they have changed recently due to a move	I want to check my shipping	C: I want to check my order.
Agent	I would be happy to help you with that	I can help you with that	
Agent	Is there an outstanding order?	How long have you been waiting?	
Agent	Or is this just an update to your account?	I have pulled up your account.	
Customer	Yes my order id is 4870952797	My order/account ID is _____	
Agent	What is your name please?	What is your name?	
Customer	Crystal Minh	<name>	A: Can I have your account/order?
Agent	What is the shipping status of the order?	What is the shipping status?	C: My account/order is _____
Customer	In Transit	In store/ In transit	
Agent	Next I need to validate your purchase. I will need your username and email.	I need your name	
Customer	cminh948, cminh948@email.com	<email>	
Agent	Thank you	Thank you	
Agent	and the new address please?	Can you tell me _____?	A: Can I have the address?
Customer	9756 Primrose Street Newark, MI 85971	<address>	C: My address is _____
Agent	All taken care of!	Your order has been updated	
Agent	Is there anything else today?	Anything else?	A: Anything else I can help?
Customer	Thank you that is all	That's all. Thank you.	C: That's all. Thank you.
Agent	Have a great one!	Have a good one!	

Table 4: An example of ABCD with cluster and state summaries. A and C stand for agent and customer, respectively.

model for unsupervised alignment of privacy policy documents. Both used unigram observation models. HMM-based conversation stages are combined with a topic-based segmentation by Chen and Yang (2020) for dialogue summarization. The use of unigram and bigram word models emphasizes topic in segmenting conversations. Our work differs in that the automatically learned speech acts are observations of the HMM, since word distributions are captured by the HT.

Most similar to our work is (Zhou et al., 2020), which uses two finite state transducers (FSTs) to map a sequence of dialogue acts (or strategies) to a sequence of state embeddings, which are then integrated into a hierarchical encoder-decoder model for prediction of the next strategy in a negotiation dialogue. The FSTs are analogous to our HMM, but the inputs are based on learning from hand-labeled strategies and rule-based dialogue acts.

There are other approaches to modeling conversation structure that do not rely on HMMs. DIALOGRAPH (Joshi et al., 2021) uses a graph attention network to encode discrete DA and strategy label sequences. A variational recurrent neural network is used to model structure by Shi et al. (2019). These approaches are less amenable to the interpretation methods used in our work.

Two key differences in our approach compared to all these studies are: i) the use of HMM topology learning via successive state splitting, and ii) the integration of structural information using a multi-stream neural sequence model.

6 Conclusion

In summary, this work combines two simple approaches for unsupervised learning on top of embedded utterance representations (K-means clustering and HMM topology design) to derive a hierarchical representation of conversation structure, which is useful to enhance a hierarchical trans-

former in three conversation-level classification tasks. The K-means clusters are intended to approximate DAs and the HMM is intended to learn sub-dialogue structure. Unlike prior work in this area, the sub-dialogues build on DA sequences rather than unigram/bigram statistics, and the HMM incorporates forward-moving dialogue flow constraints in topology learning, with the goal of capturing sub-dialogue function.

Acknowledgments

We thank Mourad Heddaya for exploring preliminary experiments when he was at the University of Washington. We also thank all members in TIAL lab and NLP groups at the University of Washington who provided valuable feedback and insights to this work.

Limitations

First, our experiments explore only two types of dialogues (negotiation and customer support) with conversation-level tasks (identifying the topic or assessing some measure of conversation success). Although THETA shows promising results, it requires further exploration with other types of conversations (e.g. information gathering, tutoring), including more examples of spoken interactions, as well as extending THETA to multi-party discussions. In addition, it would be of interest to assess the utility of automatically learned structure for other types of tasks, such as call center analytics or state tracking to support dialogue management or online agent support.

Second, we use K-means and HMMs for deriving the conversation structure, both of which require dataset-specific hyperparameters that are unlikely to transfer well to new datasets. Additionally, we only study a late fusion strategy for combining discrete structure and text-based representations. A more tightly integrated approach might be more effective. For example, our K-means DA is based on a single utterance; however, sequence models have been important for past work on unsupervised learning of DAs. Future work could leverage sequential DA dependencies in joint DA and sub-dialogue structure learning or explore continuous DA-like representations, as in (Cheng et al., 2019).

Ethical Considerations

The automatic learning of conversation structure is dependent on having data that is matched to the task of interest. A potential challenge is that biases in the data could result in some conversation strategies not being well represented. The summarization approach provides interpretability of the model, but imperfect summarizations could lead to incorrect interpretations.

References

Tim Althoff, Kevin Clark, and Jure Leskovec. 2016. Large-scale analysis of counseling conversations: An application of natural language processing to mental health. *Transactions of the Association for Computational Linguistics*, 4:463–476.

Srinivas Bangalore, Giuseppe Di Fabbrizio, and Amanda Stent. 2008. Learning the structure of task-driven human–human dialogs. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1249–1259.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 113–120, Boston, Massachusetts, USA. Association for Computational Linguistics.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics.

Florian Boudin and Emmanuel Morin. 2013. Keyphrase extraction for N-best reranking in multi-sentence compression. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 298–305, Atlanta, Georgia. Association for Computational Linguistics.

T. Brychcin and P. Kral. 2017. Unsupervised dialogue act induction using Gaussian mixtures. In *Prof. EMNLP*, volume 2, page 485–490.

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3002–3017, Online. Association for Computational Linguistics.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of the 2020 Conference on Empirical Methods in*

Natural Language Processing (EMNLP), pages 4106–4118, Online. Association for Computational Linguistics.

Hao Cheng, Hao Fang, and Mari Ostendorf. 2019. A dynamic speaker model for conversational interactions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2772–2785, Minneapolis, Minnesota. Association for Computational Linguistics.

Mark G Core and James F Allen. 1997. Coding dialogs with the damsl annotation scheme. In *in Proc. Working Notes AAAI Fall Symp. Commun. Action in Humans*.

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

Bradley Efron and Robert Tibshirani. 1993. An introduction to the bootstrap. In *Chapman & Hall/CRC Monographs on Statistics & Applied Probability*. Taylor & Francis.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

R. Chulaka Gunasekara, David Nahamoo, Lazaros C. Polymenakos, David Echeverría Ciaurri, Jatin Ganhotra, and Kshitij P. Fadnis. 2019. Quantized dialog – a general approach for conversational systems. *Computer Speech and Language*, 54:17–30.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Rishabh Joshi, Vidhisha Balachandran, Shikhar Vashishth, Alan Black, and Yulia Tsvetkov. 2021. DialoGraph: Incorporating interpretable strategy-graph networks into negotiation dialogues. In *International Conference on Learning Representations*.

Shafiq Joty, Giuseppe Carenini, and Chin-Yew Lin. 2011. Unsupervised modeling of dialog acts in asynchronous conversations. In *Proc. International Joint Conference on Artificial Intelligence*, pages 1807–1813.

Dan Jurafsky, Elizabeth Shriberg, , and Debra Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical report, University of Colorado, Boulder.

Bernard Merialdo. 1994. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155–171.

Kevin Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. MIT Press.

Mari Ostendorf and Harald Singer. 1997. HMM topology design using maximum likelihood successive state splitting. *Computer Speech & Language*, 11(1):17–41.

Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844. IEEE.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Michael J. Paul. 2012. Mixed membership Markov models for unsupervised conversation modeling. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 94–104, Jeju Island, Korea. Association for Computational Linguistics.

Rohan Ramanath, Fei Liu, Norman Sadeh, and Noah A. Smith. 2014. Unsupervised alignment of privacy policies using hidden Markov models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 605–610, Baltimore, Maryland. Association for Computational Linguistics.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over

100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of Twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180, Los Angeles, California. Association for Computational Linguistics.

Bishal Santra, Potnuru Anusha, and Pawan Goyal. 2021. Hierarchical transformer for task oriented dialog systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5649–5658, Online. Association for Computational Linguistics.

Jacob Schreiber. 2018. Pomegranate: fast and flexible probabilistic modeling in python. *Journal of Machine Learning Research*, 18(164):1–6.

Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 664–674, Melbourne, Australia. Association for Computational Linguistics.

Weiyan Shi, Tiancheng Zhao, and Zhou Yu. 2019. Unsupervised dialog structure learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1797–1807, Minneapolis, Minnesota. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

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

Ke Zhai and Jason D. Williams. 2014. Discovering latent structure in task-oriented dialogues. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–46, Baltimore, Maryland. Association for Computational Linguistics.

Yiheng Zhou, Yulia Tsvetkov, Alan W Black, and Zhou Yu. 2020. Augmenting non-collaborative dialog systems with explicit semantic and strategic dialog history. In *International Conference on Learning Representations*.

A Experimental Setup Details

We pretrain and finetune on BERT (Devlin et al., 2019) downloaded from Huggingface Transformers (Wolf et al., 2020)[3] and use the uncased base model of BERT in most of our experiments. To feed lengthy conversations to the model, we employ gradient checkpoint and DeepSpeed (Rasley et al., 2020), a deep learning optimization library, to reduce GPU memory usage and accelerate the training process.

The details of the model hyperparameters are as follows. 1-layer and 2-head transformers with 300 hidden size are applied to encode sequences of utterance-level embeddings in text view and sequences of clusters and states in structure view. Thus, the total number of parameters of our best system THETA, including base model of BERT and 3 one-layer transformers, is about 113M. For in-domain adaptation pretraining (DAPT), we use 5×10^{-5} as learning rate and 5000 steps for CRAIGSLISTBARGAIN and ABCD and 30000 steps for CALL CENTER. 0.1 epochs are used as warm-up steps with linear learning rate decay. Gradient accumulation and PyTorch (Paszke et al., 2019) distributed data parallel GPU training are applied to achieve the equivalent training batch size 4096. For finetuning, we set 1×10^{-5} as the learning rates, 4 epochs in total and 0.1 epochs for warm-up steps with linear decay. The equivalent training batch size is 16 during finetuning. Besides, the layer-wise learning rate decay is utilized to stabilize the training results; the rates are from $0.7, 0.8, 0.9$ and the 0.9 leads to the best performance. For the rest of the training hyperparameters, we follow the default values in HuggingFace's training script.

For K-means, we use Faiss (Johnson et al., 2019)[4] with GPU to speed up clustering process for large private corpus. For HMMs, we develop our splitting algorithm via Pomegranate (Schreiber, 2018),[5] a Python package that implements fast and flexible probabilistic models, to build our topology learning algorithm. The predefined numbers of clusters vary for different datasets. To compare with handcrafted DAs provided in CRAIGSLISTBARGAIN, we define number of clusters $k = 14$ for each party. For customer service domain, we set $k = 60$ for ABCD and $k = 120$ for CALL CENTER. For all datasets, we try the number of states from 5 to 20 and find the best numbers of states are 8, 12, and 12 for CRAIGSLISTBARGAIN, ABCD, and CALL CENTER, respectively. Each training run takes at most 2 hours on 2 Nvidia GeForce RTX 2080Ti GPUs for CRAIGSLISTBARGAIN and ABCD and 54 hours on 8 GPUs on CALL CENTER. All models are saved based on the best performance on the development sets. For each experiment on CRAIGSLISTBARGAIN and ABCD, we conduct 15 random runs and report the median and standard deviation. Due to the computation limitations and the size of corpus, we only conduct a single run for CALL CENTER for each experiment setting. The total number of GPU hours for all experiments, including different runs with random seeds, is 1536 hours approximately.

	CRAIGSLIST BARGAIN	ABCD	CALL CENTER
# dialogues	6682	10042	949410
# turns / dialogue	9.2	22.1	71.6
# tokens / turn	15.5	9.2	16.3
# tokens / dialogue	142.6	202.5	1167.1

Table 5: Data statistics of the datasets.

	CRAIGSLIST BARGAIN	ABCD	CALL CENTER
train set # dialogues	4828	8034	711310
dev. set # dialogues	561	1004	95540
test set # dialogues	567	1004	142560

Table 6: Train/dev./test split of datasets

B Dataset Details

We follow all original data preprocessing scripts for CRAIGSLISTBARGAIN[6] and ABCD.[7] For the private collection of customer service conversations, CALL CENTER, all private user information is anonymized. The data statistics are summarized in Table 5 and Table 6.

C Topology with Summaries

Figure 4 shows the detailed topology with both cluster and sub-dialogue state summaries. For each sub-dialogue state, we add the cluster summaries with

[3] https://github.com/huggingface/transformers

[4] https://github.com/facebookresearch/faiss

[5] https://github.com/jmschrei/pomegranate

[6] https://github.com/rishabhjoshi/DialoGraph_ICLR21.

[7] https://github.com/asappresearch/abcd.

top 3 emission probabilites and the sub-dialogue state summaries for the buyer and the seller. The thickness of edges indicates the levels of negotiation success and the edges with probabilities lower than 0.01 are pruned for simplicity.

D License of Artifacts

The license of code for Wolf et al. (2020) and Schreiber (2018) are Apache license version 2.0. The license of code for Joshi et al. (2021), Rasley et al. (2020), and Chen et al. (2021) are MIT License. The terms for use of our artifacts will be included in our released package.

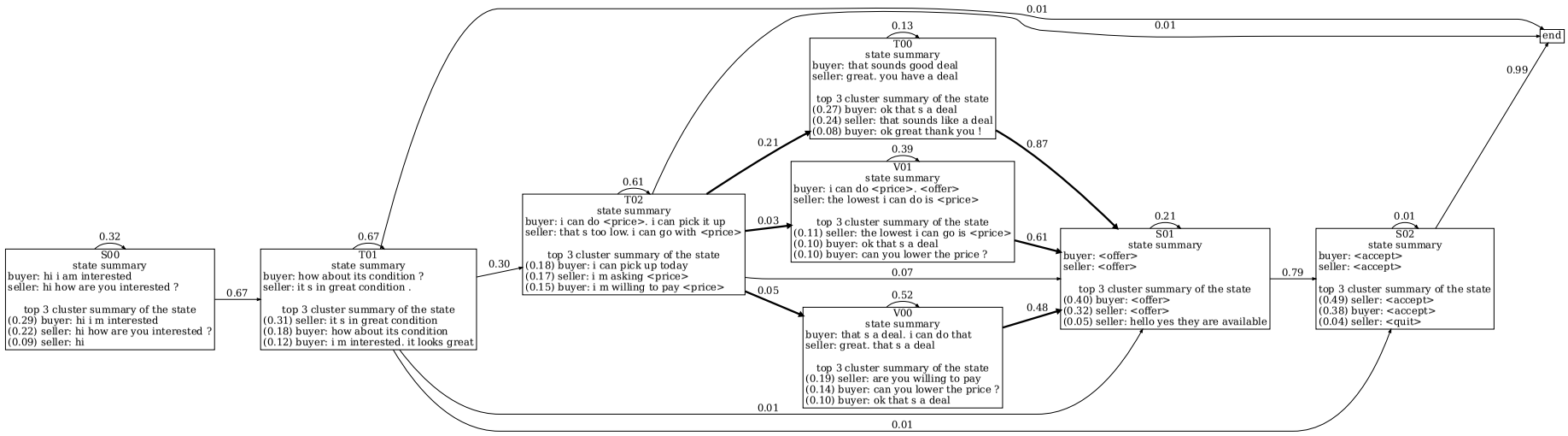


Figure 4: The 8-state full topology with cluster and sub-dialogue state summaries on CRAIGSLISTBARGAIN dataset. The thicker edges represent higher levels of negotiation success.