A Dynamic Speaker Model for Conversational Interactions

Hao Cheng **Hao Fang** **Mari Ostendorf**
University of Washington
{chenghao,hfang,ostendorf}@uw.edu

Abstract

Individual differences in speakers are reflected in their language use as well as in their interests and opinions. Characterizing these differences can be useful in human-computer interaction, as well as analysis of human-human conversations. In this work, we introduce a neural model for learning a dynamically updated speaker embedding in a conversational context. Initial model training is unsupervised, using context-sensitive language generation as an objective, with the context being the conversation history. Further fine-tuning can leverage task-dependent supervised training. The learned neural representation of speakers is shown to be useful for content ranking in a socialbot and dialog act prediction in human-human conversations.[1]

1 Introduction

Representing language in context is key to improving natural language processing (NLP). There are a variety of useful contexts, including word history, related documents, author/speaker information, social context, knowledge graphs, visual or situational grounding, etc. This paper addresses the problem of modeling the speaker. Accounting for author/speaker variations has been shown to be useful in many NLP tasks, including language understanding (Hovy and Søgaard, 2015; Volkova et al., 2013), language generation (Mirkin et al., 2015; Li et al., 2016), human-computer dialog policy (Bowden et al., 2018), query completion (Jaech and Ostendorf, 2018; Shokouhi, 2013), comment recommendation (Agarwal et al., 2011) and more. In this work, we specifically focus on dialogs, including both human-computer (socialbot) and human-human conversations.

While many studies rely only on discrete metadata and/or demographic information, such information is not always available. Thus, it is of interest to learn about the speaker from the language directly, as it relates to the person's interests and speaking style. Motivated by the success of unsupervised contextualized representation learning for words and documents (Mikolov et al., 2013; Kiros et al., 2015; McCann et al., 2017; Peters et al., 2018; Devlin et al., 2019), our approach is to use unsupervised learning with a neural model of a speaker's dialog history. The model uses latent speaker mode vectors for representing a speaker turn as in (Cheng et al., 2017), which provides a framework for analysis of what the model learns about speaking style. Further, the model is structured to allow a dynamic update of the speaker vector at each turn in a dialog, in order to capture changes over time and improve the speaker representation with added data.

The speaker embeddings can be used as context in conversational language understanding tasks, e.g., as an additional input in dialog policy prediction in human-computer dialogs or in understanding dialog acts in human-human dialogs. In the supervised training of such tasks, the speaker model can be fine-tuned.

This work makes two primary contributions. First, we propose a neural model for learning dynamically updated speaker embeddings in conversational interactions. The model training is unsupervised, relying on only the speaker's conversation history rather than meta information (e.g., age, gender) or audio signals which may not be available in a privacy-sensitive situation. The model also has a learnable component for analyzing the latent modes of the speaker, which can be helpful for aligning the learned characteristics of a speaker with the human-interpretable factors. Second, we use the learned dynamic speaker embed-

[1] The implementation of code is available at https://github.com/hao-cheng/dynamic_speaker_model.git

Proceedings of NAACL-HLT 2019, pages 2772–2785
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

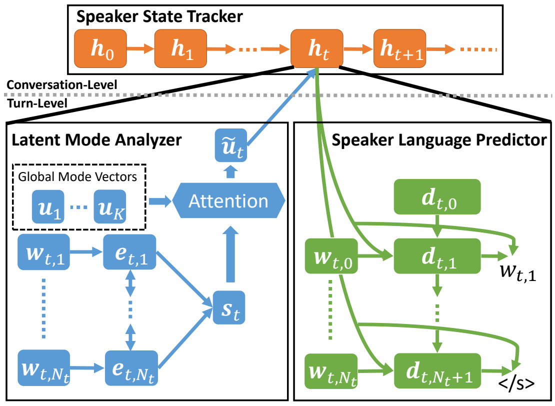


Figure 1: The dynamic speaker model. The speaker state tracker operates at the conversation level. The latent model analyzer and speaker language predictor operate at the turn level. The figure only shows processes in those two components for the turn t.

dings in two representative tasks in dialogs: predicting user topic decisions in socialbot dialogs, and classifying dialog acts in human-human dialogs. Empirical results show that using the dynamic speaker embeddings significantly outperforms the baselines in both tasks. In the public dialog act classification task, the proposed model achieves the state-of-the-art results.

2 Dynamic Speaker Model

In this section, we start with an overview of the proposed model for learning speaker embeddings that are dynamically refined over the course of a conversation. Details about individual components are described in subsequent subsections.

The model is based on two motivations. First, a speaker's utterances reflect intents, speaking style, etc. Thus, we may build speaker embeddings by analyzing latent modes that characterize utterances in terms of such characteristics, apart from topic-related interests a user might have. Second, the information about a speaker is accumulated as the conversation evolves, which allows us to gradually refine and update the speaker embeddings. The speaker embeddings can be directly used as features or fine-tuned based on the downstream tasks. We design the dynamic speaker model to focus on learning cues from the speaker's utterances, and leave the modeling of different speaker-addressee interactions for supervised downstream tasks.

The model consists of three components as illustrated in Fig. 1. First, a **latent mode analyzer** reads in an utterance and analyzes its latent modes. It processes the speaker's turns independently of each other and builds a local speaker mode vector for each turn. To accumulate speaker information as the conversation evolves, we build a **speaker state tracker** that maintains speaker states at individual turns. At each turn, it takes two input vectors to update the speaker state: 1) the local speaker mode vector for the current turn from the latent mode analyzer, and 2) the speaker state at the previous turn from the tracker itself. Finally, we employ a **speaker language predictor** to drive the learning of the latent model analyzer and the speaker state tracker. It reconstructs the utterance using the corresponding speaker state. Intuitively, the speaker language predictor models overall linguistic regularities itself and uses the speaker state to supply information related to speaker characteristics. For sequence modeling in all three components, we use the long short-term memory (LSTM) recurrent neural network (Hochreiter and Schmidhuber, 1997). In our experiments, the three components are trained jointly.

2.1 Latent Mode Analyzer

At each turn t, the latent mode analyzer constructs a local speaker mode vector $\tilde{\mathbf{u}}_t \in \mathbb{R}^c$ that captures salient characteristics of the speaker's current utterance for use in the dynamic speaker model. First, the utterance word sequence $w_{t,1}, \cdots, w_{t,N_t}$ is mapped to an embedding sequence, where $w_{t,n}$ is represented with $\mathbf{w}_{t,n} \in \mathbb{R}^d$ according a lookup with dictionary $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}|\times d}$ associated with vocabulary $\mathcal{V}$. Then, the latent mode analyzer goes through two stages to construct $\tilde{\mathbf{u}}_t$.

In the first stage, a bi-directional LSTM (Bi-LSTM), which consists of a forward LSTM and a backward LSTM, is used to encode the word embedding sequence into a fixed-size utterance summary vector $\mathbf{s}_t \in \mathbb{R}^{2m}$, where m is the dimension of the hidden layer in the forward and backward LSTMs. Formally, the forward LSTM computes its hidden states as $\mathbf{e}^F_{t,n} = g^F(\mathbf{w}_{t,n}, \mathbf{e}^F_{t,n-1}) \in \mathbb{R}^m$ for $n = 1, \ldots, N_t$, where $g^F(\cdot, \cdot)$ denotes the forward LSTM function. The backward LSTM computes its hidden states $\mathbf{e}^B_{t,n} \in \mathbb{R}^m$ similarly. The initial hidden states $\mathbf{e}^F_{t,0}$ and $\mathbf{e}^B_{t,N_t+1}$ are set to zeros. The summary vector $\mathbf{s}_t$ is the concatenation of the two final hidden states, $\mathbf{s}_t = [\mathbf{e}^F_{t,N_t}, \mathbf{e}^B_{t,1}]$.

In the second stage, the utterance summary vector $\mathbf{s}_t$ is compared with K global mode vectors

$\mathbf{u}_1, \ldots, \mathbf{u}_K \in \mathbb{R}^c$ which are learned as part of the model. The association score $a_{t,k}$ between $\mathbf{s}_t$ and $\mathbf{u}_k$ is computed using the dot-product attention mechanism (Vaswani et al., 2017) as follows,

$$a_{t,k} = \frac{\exp(\langle \mathbf{P}\mathbf{s}_t, \mathbf{Q}\mathbf{u}_k \rangle)}{\sum_{k'=1}^{K} \exp(\langle \mathbf{P}\mathbf{s}_t, \mathbf{Q}\mathbf{u}_{k'} \rangle)}, \quad (1)$$

where $\mathbf{P} \in \mathbb{R}^{c \times 2m}$ and $\mathbf{Q} \in \mathbb{R}^{c \times c}$ are learnable weights, and $\langle \cdot, \cdot \rangle$ indicates the dot-product of two vectors. The local speaker mode vector is then constructed as $\tilde{\mathbf{u}}_t = \sum_{k=1}^{K} a_{t,k} \mathbf{u}_k$.

2.2 Speaker State Tracker

The speaker state tracker provides a dynamic summary of speaker language features observed in the conversation history, using an LSTM to encode the sequence of local speaker mode vectors $\tilde{\mathbf{u}}_{t,1}, \cdots, \tilde{\mathbf{u}}_{t,N_t}$. At turn t, this LSTM updates its hidden state $\mathbf{h}_t \in \mathbb{R}^m$ using the local speaker mode vector $\tilde{\mathbf{u}}_t$ and its previous hidden state $\mathbf{h}_{t-1} \in \mathbb{R}^m$, i.e., $\mathbf{h}_t = g^S(\tilde{\mathbf{u}}_t, \mathbf{h}_{t-1})$, where $g^S(\cdot, \cdot)$ is the speaker LSTM function. The hidden state $\mathbf{h}_t$ provides the speaker state vector at turn t.

2.3 Speaker Language Predictor

The speaker language predictor is a conditional LSTM language model (LM) that reconstructs the word sequence in the current turn. Language modeling is a way to provide a training signal for unsupervised learning that models the conditional probability $\Pr(w_{t,n}|w_{t,<n})$, where $w_{t,<n}$ denotes all preceding words of $w_{t,n}$ in the turn t.

The speaker language predictor uses the same dictionary $\mathbf{W}$ for word embeddings as the latent mode analyzer to represent words at time t. The initial hidden state $\mathbf{d}_{t,0} \in \mathbb{R}^m$ of the LSTM is set to $\tanh(\mathbf{L}\mathbf{h}_t)$, where $\mathbf{L} \in \mathbb{R}^{m \times m}$ is a learnable matrix and $\tanh(\cdot)$ is the hyperbolic tangent function. Subsequent LSTM hidden states are computed as

$$\mathbf{d}_{t,n} = g^{LM}(r^I(\mathbf{w}_{t,n-1}, \mathbf{h}_t), \mathbf{d}_{t,n-1}),$$

for $n = 1, \ldots, N_t + 1$, where $r^I(\mathbf{w}_{t,n-1}, \mathbf{h}_t) = \mathbf{R}_w^I \mathbf{w}_{t,n-1} + \mathbf{R}_h^I \mathbf{h}_t$ is a linear transformation with learned parameters $\mathbf{R}_w^I \in \mathbb{R}^{m \times d}$ and $\mathbf{R}_h^I \in \mathbb{R}^{m \times m}$, $g^{LM}(\cdot, \cdot)$ is a forward LSTM function, and $\mathbf{w}_{t,0}$ is the word embedding for the start-of-sentence token. By injecting the speaker state vector at every time step n in the turn t, the model is more likely to favor directly using the speaker state vector (vs. the word history) for predicting the speaker language. The conditional probability is then computed as

$$\Pr(w_{t,n}|w_{t,<n}) = \text{softmax}(\mathbf{V} r^O(\mathbf{h}_t, \mathbf{d}_{t,n})), \quad (2)$$

where $\mathbf{V} \in \mathbb{R}^{|\mathcal{V}| \times m}$ is the weight matrix, and $r^O(\mathbf{h}_t, \mathbf{d}_{t,n}) = \mathbf{R}_h^O \mathbf{h}_t + \mathbf{R}_d^O \mathbf{d}_{t,n}$ is another linear function with learnable parameters $\mathbf{R}_h^O, \mathbf{R}_d^O \in \mathbb{R}^{m \times m}$. The last word w_{t,N_t+1} is always the end-of-sentence token.

2.4 Model Training and Tuning

The training objective for a given conversation is the log-likelihood $\sum_t \sum_n \log \Pr(w_{t,n}|w_{t,<n})$, where the conditional probability is defined in (2). The Adam optimizer (Kingma and Ba, 2015) is used with a configuration of $\beta_1 = 0.9$ and $\beta_2 = 0.97$. The initial learning rate is set to 0.002. We halve the learning rate at each epoch once the development log-likelihood decreases, and terminate the training when it decreases for the second time. This validation protocol is used throughout the paper for training the proposed model.

In our experiments, the embedding dictionary $\mathbf{W}$ is initialized using pre-trained 300-dimensional word embeddings (Bojanowski et al., 2017) for words within the vocabulary of this resource. The remaining part of $\mathbf{W}$ and other model parameters are randomly initialized based on $\mathcal{N}(0, 0.01)$. The mode vector dimension c is set to 64. We tune the number of global mode vectors K from $\{16, 32\}$ and the hidden layer size m from $\{128, 160\}$. The final model is selected based on the log-likelihood on the development set.

3 User Topic Decision Prediction

We first study a prediction task that estimates whether the user engaged in a socialbot conversation would accept a suggested topic. Specifically, we use a corpus of human-socialbot conversations collected during the 2017 Alexa Prize competition (Ram et al., 2017) from the Sounding Board system (Fang et al., 2018; Fang, 2019). Due to privacy concerns, the socialbot does not have access to any identity information about users. Also, since each device may be used by multiple users, the device address is not a reliable indicator of the user ID. Therefore, the ability to profile the user through one conversational interaction is desirable for guiding the socialbot's dialog policy.

	train	dev	test
# conversations	19,076	6,321	6,465
# topic decisions	85,340	28,060	29,561

Table 1: Data statistics of the topic decision dataset.

3.1 Data

Each conversation begins with a greeting and ends when the user makes a stop command. The socialbot engages the user in the conversation using a wide range of content indexed by topics, where a topic corresponds to a noun or noun phrase that refers to a named entity (e.g., Google) or a concept (e.g., artificial intelligence). These topics are extracted using both constituency parsing results of the textual content and content meta-information. During the conversation, the socialbot sometimes negotiates the topic with the user using an explicit confirmation turn and records the user's binary decision (accept or reject) on the topic.

In socialbot conversations, a system turn is always followed by a user turn and vice versa. We tag system turns making explicit confirmation about a topic and attach the corresponding binary user decisions with them. To curate the dataset for the topic decision prediction task, we use a total of 31,862 conversations with more than 5 user turns. On average there are around 22 user turns per conversation. Not every system turn makes a topic suggestion, and the average number of topic decisions per conversation is 4.5. We randomly split the conversations into training, development, and test sets by $3/1/1$. The data statistics are shown in Table 1. In our experiments, we directly use the speech recognition output of user utterances. The vocabulary $\mathcal{V}$ consists of roughly 11K words that appear at least 5 times in the training set.

3.2 Topic Decision Classifier

We use a feed-forward neural network (FFNN) to make binary predictions (accept vs. reject) for individual topic suggestions. For each topic suggestion, the FFNN takes two inputs: 1) an embedding $\mathbf{x}_{t'}$ for the suggested topic at system turn t', and 2) a user embedding vector $\mathbf{z}_t$ at user turn t. Note the model does not have information about user turns after the system turn t' when making the prediction, i.e., the user turn t appears before the system turn t'.

The topic embedding $\mathbf{x}_{t'}$'s are looked up from the embedding dictionary learned by the FFNN. They are initialized by averaging the embeddings of their component words using the public pre-trained 300-dimensional word embeddings (Bojanowski et al., 2017).

For the user embedding vector, we explore two settings that use different numbers of user turns as context. In both settings, topic decisions occurring in the first 5 user turns are not used for evaluations.

Static User Embeddings: Motivated by the findings that most user characteristics can be inferred from initial interactions (Ravichander and Black, 2018), we derive a static user embedding vector for a conversation using the first 5 user turns and apply it for predicting topic decisions afterwards.

Dynamic User Embeddings: Alternatively, we build a user embedding vector for user turn t using all previous user turns. Here, a topic decision for system turn t' is aligned with its preceding user turn t.

In our experiments, we compare different unsupervised models with our proposed dynamic speaker model. For both settings, all unsupervised models are pre-trained on *all* user turns in training conversations. They are fixed when training the FFNN classifier. The FFNN classifier is trained with the logistic loss using the Adam optimizer (Kingma and Ba, 2015). The training protocol is similar to that described in §2.4. We tune the hidden layer size from $\{64, 128\}$ and the number of hidden layers from $\{0, 1\}$. The model is selected based on the loss on the development set.

In addition, we use a user-agnostic **TopicPrior** baseline. It builds a probability lookup for each topic using its acceptance rate on the training set. We tune a universal probability threshold for all topics based on the development set accuracy.

In all experiments, three evaluation metrics are used: accuracy, area under the receiver operating characteristic curve (AUC), and normalized cross-entropy (N-CE). N-CE is computed as the relative cross-entropy reduction of the model over the TopicPrior baseline.

3.3 Experiments: Static User Embeddings

As described in §3.2, we use the first 5 user turns to derive the user embedding vector for a conversation. We compare our dynamic speaker model with three other unsupervised models.

DynamicSpeakerModel: For the proposed dynamic speaker model, we concatenate the speaker state vector $\mathbf{h}_t$ and the local speaker mode vector

Model	Acc	AUC	N-CE
TopicPrior	68.8	72.5	0
UtteranceLDA	68.8	73.1	12.6
UtteranceAE	68.8	73.4	12.8
TopicDecisionEncoder	68.9	73.8	13.4
DynamicSpeakerModel	**69.5**	**74.2**	**13.7**

Table 2: Test set results (in %) for topic decision predictions using *static* user embeddings.

$\tilde{\mathbf{u}}_t$ for each of the first 5 user turns. Then, we apply the max-pooling operation over the 5 concatenated vectors to summarize all the information. The resulting vector $\tilde{\mathbf{h}}$ is used as the user embedding vector.

UtteranceLDA: The latent Dirichlet allocation (LDA) model (Blei et al., 2003) is trained with 16 latent groups by treating all user utterances in a conversation as a document.[2] The trained LDA model builds a 16-dimensional probability vector as the user embedding vector by loading the first 5 user turns as a single document.

UtteranceAE: The utterance auto-encoder model is built upon the sequence auto-encoder (Dai and Le, 2015). We replace the original encoder by a BiLSTM that encodes the utterance at user turn t into a summary vector $\mathbf{s}_t$ in the same way as the first stage of the latent mode analyzer described in §2.1. The auto-encoder is trained on all user utterances in the training data, using the same training protocol described in §2.4. We set the hidden layer size to 128. The user embedding vector is constructed by applying the max-pooling operation over the summary vectors $\mathbf{s}_1, \ldots, \mathbf{s}_5$ for the first 5 user turns.

TopicDecisionEncoder: This model encodes the topic decisions occurred in the first 5 user turns. The user embedding vector is the concatenation of two vectors. One is max-pooled from the topic embeddings for accepted topics, and the other for rejected topics, both include a dummy topic vector as default. The topic embeddings are composed by averaging the public pre-trained 300-dimensional embeddings (Bojanowski et al., 2017) for words in the topic.

Experiment results are summarized in Table 2. The TopicPrior is a very strong predictor, with an accuracy on par with other user embeddings. This indicates that the popularity-based approach is a good start for content ranking in socialbots when there is little user information. Nevertheless, we can still observe some improvement over the TopicPrior in terms of AUC and N-CE, which suggests using information from initial interactions reduces the uncertainty of predictions. The proposed dynamic speaker model performs the best among the compared models, reducing the cross-entropy by 13.7% over the TopicPrior baseline.

Model	Acc	AUC	N-CE
TopicDecisionLSTM	69.3	74.8	14.6
UtteranceAE + LSTM	69.9	75.4	15.3
DynamicSpeakerModel	**72.4**	**79.0**	**20.0***

Table 3: Test set results (in %) for topic decision predictions using *dynamic* user embeddings. *: The improvement of DynamicSpeakerModel over both TopicDecisionLSTM and UtteranceAE + LSTM is statistically significant based on both t-test and McNemar's test ($p < .001$).

3.4 Experiments: Dynamic User Embeddings

Here, we use all information accumulated before the system turn of suggesting the topic to build the corresponding user embedding vector. Since the UtteranceLDA is not as effective based on static embedding experiments, we only consider extending UtteranceAE and TopicDecisionEncoder models for comparison here.

DynamicSpeakerModel: The speaker state tracker in our model accumulates the user information as the conversation evolves. Thus, we directly concatenate the speaker state vector $\mathbf{h}_t$ and the local speaker mode vector $\tilde{\mathbf{u}}_t$ as the user embedding vector at user turn t. Other than using more turns, this is the same DynamicSpeakerModel configuration as in §3.3.

UtteranceAE+LSTM: This model uses an LSTM to encode the summary vector sequence derived from the same utterance auto-encoder used in §3.3. The LSTM hidden states are treated as user embedding vectors at individual user turns.

TopicDecisionLSTM: Similarly, an LSTM is used to encode the topic decision sequence. At each time step, the LSTM reads the concatenation of the topic embedding and the one-hot vector encoding the topic decision. We use the same topic embeddings as the TopicDecisionEncoder in §3.3. Since not every user turn is associated with a topic

[2] To allow the LDA model to take into account bi-grams, we replace the uni-gram token w_i with its bi-gram (w_i, w_{i+1}) concatenated as a single token if the bi-gram is among the top 500 frequent bi-grams.

decision, the time steps of this LSTM are aligned to a sequence of non-consecutive user turns. The LSTM hidden states are treated as user embedding vectors at corresponding user turns.

For UtteranceAE+LSTM and TopicDecisionLSTM, the hidden layer size of the LSTM is set to 128. While the utterance auto-encoder and topic embeddings are pre-trained, the LSTM components are jointly learned with the FFNN for composing dynamic user embeddings.

Experiment results are shown in Table 3. The DynamicSpeakerModel performs the best. Comparing to results in Table 2, all three unsupervised models outperform their static counterparts, which suggests the advantage of using dynamic context for predicting user topic decisions as conversation evolves.

Statistical significance tests of the difference in performance of two systems were conducted under both the t-test using the predicted probabilities and McNemar's test using the binary predictions. Under both tests, the predictions from the TopicDecisionLSTM and the DynamicSpeakerModel are highly signification ($p < .001$). Predictions from UtteranceAE + LSTM and DynamicSpeakerModel are also significantly different based on both tests ($p < .001$).

3.5 Qualitative Analysis

First, we manually inspect the predictions from the TopicDecisionLSTM and DynamicSpeakerModel used in §3.4 and the static baseline TopicPrior in §3.3. Compared with TopicPrior, we find that TopicDecisionLSTM is able to utilize the semantic relatedness between neighboring topics and corresponding user decisions. For example, "Elon Musk" (the CEO) is likely to be rejected if "Tesla" (the company) has been rejected earlier, though both are popular topics with high acceptance rates. In addition, it seems that the DynamicSpeakerModel is able to make use of user reactions. In the anecdotal example illustrated in Table 4, the user accepts the topic "Arnold Schwarzenegger" which is correctly predicted by both TopicDecisionLSTM and DynamicSpeakerModel, but only the DynamicSpeakerModel correctly predicts the rejection of "politics" later.

We then analyze what language features are learned by latent modes in our dynamic speaker model. For each mode, we extract top utterances sorted by their association scores as computed in (1). Examples from the most representative modes are provided in Appendix A. In brief, we find two separate modes related to positive and negative reactions; other modes correspond to classes of dialog acts, such as yes/no answers, topic requests and conversation-closing. Within topic request modes, some involve short topic phrases (e.g., "*holidays*") while others use complete requests (e.g. "*can we talk about cats*"). Along this line, some modes are associated with relatively terse users and others with talkative users. These findings indicate that our model cpatures various user characteristics that might be useful for predicting their interaction preferences.

Bot: Do you like the actor Arnold Schwarzenegger?
User: yeah before he got into politics
Bot: Super, would you like to know a fun fact about Arnold Schwarzenegger?

- **TopicDecisionLSTM**: accept
- **DynamicSpeakerModel**: accept

User: why not sure
...
Bot: I'm running out of things to say about him. Do you wanna hear some news about politics?

- **TopicDecisionLSTM**: accept
- **DynamicSpeakerModel**: reject

User: no

Table 4: A dialog snippet showing topic decision predictions from TopicDecisionLSTM and DynamicSpeakerModel. Topics are shown with underscores.

4 Dialog Act Classification

Dialog act analysis is widely used for conversations, which identifies the illocutionary force of a speaker's utterance following the speech act theory (Austin, 1975; Searle, 1969). In this section, we apply the proposed dynamic speaker model to the dialog act classification task.

4.1 Data

We use the Switchboard Dialog Act Corpus (SwDA), which has dialog act annotations on two-party human-human speech conversations (Jurafsky et al., 1997; Stolcke et al., 2000). In total, there are 1155 open-domain conversations with manual transcripts. Following recent work, we use 1115 conversations for training, 19 for testing, and the rest 21 for development.[3] The original fine-grained dialog act labels are mapped to 42

[3] The training and test split files are downloaded from https://web.stanford.edu/~jurafsky/ws97/.

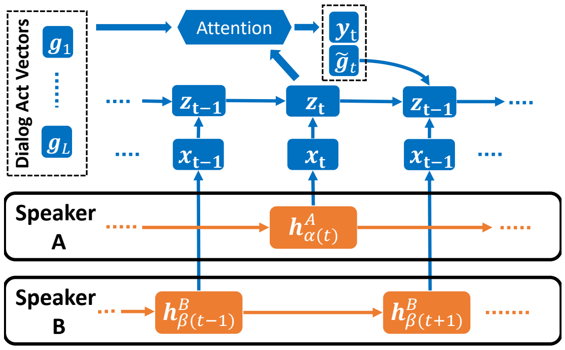


Figure 2: The attention-based LSTM tagging model for dialog act classification. The figure only shows the attention operation for turn t. The lower two boxes represent two speaker state trackers.

classes.[4] For this set of experiments, we use the golden segmentation and manual transcripts provided in the dataset.

Motivated by the recent success of unsupervised models (Peters et al., 2018; Devlin et al., 2019), we also study whether the dynamic speaker model can benefit from training on external unlabelled data. Thus, we use speech transcripts from 5850 conversations from the Fisher English Training Speech Part 1 Transcripts (Cieri et al., 2004), which (like Switchboard) consists of two-party human-to-human telephone conversations but without annotations for dialog acts.

4.2 Dialog Act Tagging Model

We use an attention-based LSTM tagging model for the dialog act classification. As shown in Fig. 2, the tagging LSTM is stacked on two speaker state trackers. Note the two trackers share the same parameters as well as the underlying latent mode analyzer and speaker language predictor. They generate speaker embeddings by tracking corresponding speakers separately.

Let $\alpha(t)$ and $\beta(t)$ denote the mappings from the global turn index t to the speaker-specific turn indices for speaker A and speaker B, respectively. The mapping returns a null value if the turn t is not associated with the corresponding speaker. The speaker state vectors are used as the input to the tagging LSTM for corresponding turns, i.e., $\mathbf{x}_t = I(\mathbf{h}^A_{\alpha(t)}, \mathbf{h}^B_{\beta(t)})$ where $I(\cdot,\cdot)$ is a switcher that chooses $\mathbf{h}^A_{\alpha(t)}$ or $\mathbf{h}^B_{\beta(t)}$ depending on whether $\alpha(t)$ and $\beta(t)$ return a non-null value.

The tagging LSTM also maintains a dictionary of L dialog act vectors $\mathbf{g}_1, \ldots, \mathbf{g}_L$. The dialog act probabilities $\mathbf{y}_t \in \mathbb{R}^L$ at turn t are computed using the dot-product attention mechanism, i.e., $\mathbf{y}_t = f(\mathbf{z}_t, [\mathbf{g}_1, \ldots, \mathbf{g}_L])$, where $f(\cdot,\cdot)$ is defined as in (1), and $\mathbf{z}_t$ is the hidden state vector of the LSTM.

The tagging LSTM computes hidden states as

$$\mathbf{z}_{t+1} = g^{DA}\left(r^{DA}(\tilde{\mathbf{g}}_t, \mathbf{x}_{t+1}), \mathbf{z}_t\right)$$

where $\tilde{\mathbf{g}}_t = \sum_{l=1}^{L} y_{t,l}\mathbf{g}_l$, $g^{DA}(\cdot,\cdot)$ is the LSTM function, and $r^{DA}(\cdot,\cdot)$ is a linear function with learnable parameters. In this way, both the history dialog act predictions and the utterance information are encoded in the hidden states.

The training objective of the tagging LSTM is the sum of the cross-entropy between the dialog act label and the probabilities $\mathbf{y}_t$ at each turn. The training configuration is the same as the topic decision classifier described in §3.2. We tune the size of hidden states $\mathbf{z}_t$ and dialog act embeddings $\mathbf{g}_l$ from $\{64, 128\}$ with arbitrary combinations, and vary the number of LSTM hidden layers from $\{1, 2\}$. The best model is selected according to the development set accuracy.

4.3 Experiment Results

In our experiments, we compare three settings for using the dynamic speaker model. In the **pre-train** setting, the dynamic speaker model is trained on the SwDA data without the dialog act labels. We then freeze the model when training the tagging LSTM. In contrast, in the **pretrain + fine-tune** setting, the dynamic speaker model is fine-tuned together with the tagging LSTM. Finally, in the **pre-train w/Fisher + fine-tune** setting, the dynamic speaker model is pre-trained on the combination of SwDA and Fisher datasets, and then fine-tuned together with the tagging LSTM on the SwDA dataset. For all three settings, we use the same vocabulary $\mathcal{V}$ of size 21K which combines all tokens from the SwDA training set and those appearing at least 5 time in the Fisher corpus.

We compare our results to best published results. In (Kalchbrenner and Blunsom, 2013), a convolutional neural network (CNN) is used to encode utterances. A recurrent neural network (RNN) is then applied on top of the CNN to encode both utterances and speaker label information for predicting the dialog acts. Ji et al. (2016) propose a discourse-aware RNN LM by treating

[4] Dialog act labels are mapped using scripts from `http://compprag.christopherpotts.net/swda.html`. Utterances labelled as "segment" are merged with corresponding previous utterance by the same speaker.

Model	Acc (%)
(Kalchbrenner and Blunsom, 2013)	73.9
(Tran et al., 2017a)	74.2
(Tran et al., 2017b)	74.5
(Tran et al., 2017c)	75.6
(Ji et al., 2016)	77.0
pre-train	75.6
pre-train + fine-tune	**77.2**
pre-train w/ Fisher + fine-tune	**78.6***

Table 5: Test set accuracy for SwDA dialog act classification. *: The improvement of pre-train w/ Fisher + fine-tune is statistically significant over pre-train + fine-tune based on McNemar's test ($p < .001$).

the dialog act as a conditional variable to the LM. Tran et al. (2017a,b,c) focus on building hierarchies of RNNs to model the dialog context using previous utterances or dialog act predictions. Results from (Lee and Dernoncourt, 2016) and (Liu et al., 2017) are not directly comparable due to different experiment settings.

Experiment results are summarized in Table 5. Our pre-train setting performs on par with previous state-of-the-art supervised models except (Ji et al., 2016). Fine-tuning significantly improves the performance and allows the model to achieve a similar accuracy as (Ji et al., 2016). The best result is achieved by pre-training the dynamic speaker model with both SwDA and Fisher datasets. The improvement of pre-train w/ Fisher + fine-tune is statistically significant over pre-train + fine-tune based on McNemar's test ($p < .001$). This illustrates the advantage of the unsupervised learning approach for the proposed model as it can exploit a large amount of unlabelled data.

4.4 Qualitative Analysis

We analyze the latent modes learned on SwDA using the same approach as in §3.5. Again, specific examples are included in Appendix A. Overall, there are several modes corresponding to coarse-grained dialog acts, such as statements, questions, agreement, backchannel and conversation-closing. Many modes characterize statements, probably due to their high relative frequency in the corpus. Among the statement modes, there are two distinct groups, one containing multiple filled pauses, such as *uh, you know, well*, and the other one with *because*-clauses. The fact that coarse-grained dialog act information is partly encoded in the modes may be helping with recognizing the dialog act.

In addition, we use the speaker gender information available in the SwDA data to determine whether the latent modes in the dynamic speaker model pick up gender-related language variation. Specific examples and statistics are included in Appendix B. The Cohen-d score (Cohen, 1988) is used to measure the strength of the difference between association score distributions of male vs. female utterances for individual modes. Based on the Cohen-d score, we identified two modes that have a strong association with male speakers, and two with female speakers. All have significantly different ($p < 0.001$) distributions of association scores for female vs. male speakers using Mann-Whitney U test. In the top associated utterances for the male modes, we find utterances with several filled pauses, which has been found to be indicative of male speakers in previous work on Switchboard (Boulis and Ostendorf, 2005). The female modes are mostly agreement, acknowledgement and backchannel, which aligns with a popular sociolinguistic theory that females are more responsive (Coates, 1998). Based on this, we conclude that some speaker gender language variations are indeed captured by the learned modes.

5 Related Work

As reviewed by Zukerman and Litman (2001), user modeling for conversational systems has a long history. The research can be tracked back to the GRUNDY system (Rich, 1979) which categorizes users in terms of hand-crafted sets of user properties for book recommendation. Other systems have focused on different aspects of users, e.g., the expertise level of the user in a specific domain (Chin, 1986; Sleeman, 1985; Paris, 1987; Hovy, 1987), the user's intent and plan (Allen and Perrault, 1980; Carberry, 1983; Litman, 1986; Moore and Paris, 1992), and the user's personality (Mairesse and Walker, 2006; DeVault et al., 2014; Fung et al., 2016; Fang et al., 2017). User modeling has also been employed for personalized topic suggestion in recent Alexa Prize socialbots, using a pre-defined mapping between personality types and topics (Fang et al., 2017), or a conditional random field sequence model with hand-crafted user and context features (Ahmadvand et al., 2018). Modeling speakers with continuous embeddings for neural conversation models is studied in (Li et al., 2016), where the model directly learns a dictionary of speaker embeddings. Our unsupervised dynamic speaker model differs from previous work in that we build speaker embeddings as a weighted combination of latent modes with weights computed based on the utterance. Thus, the model can construct embeddings for any new users and dynamically update the embeddings as the conversation evolves.

Speaker language variances have been analyzed by previous work and incorporated in NLP models. Preoţiuc-Pietro et al. (2016) and Johannsen et al. (2015) find that speaker-level language variance affects lexical choices and even syntactic structure based on psycholinguistic hypotheses. Speaker demographics are used to improve both low-level tasks such as part-of-speech tagging (Hovy and Søgaard, 2015) and high-level applications such as sentiment analysis (Volkova et al., 2013) and machine translation (Mirkin et al., 2015). Lynn et al. (2017) introduce a continuous adaptation method to include user age, gender, personality traits and language features for personalizing several supervised NLP models. Different from previous work, we study the use of speaker embeddings learned from utterances in an unsupervised fashion and analyze the possible interpretability of the latent modes.

6 Conclusion

In this paper, we address the problem of modeling speakers from their language using an unsupervised approach. A dynamic speaker model is proposed to learn speaker embeddings that are updated as the conversation evolves. The model achieves promising results on two representative tasks in dialogs: user topic decision prediction in human-socialbot conversations and dialog act classification in human-human conversations. In particular, we demonstrate that the model can benefit from unlabelled data in the dialog act classification task, where we achieve the state-of-the-art results. Finally, we carry out analysis on the learned latent modes on both tasks, and find cues that suggest the model captures speaker characteristics such as intent, speaking style, and gender. For future work, it could be interesting to explore guiding some latent modes with a few examples to pick up specific user features such as personality traits.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback. We also thank Trang Tran for her feedback on the paper. This research is supported by Amazon Alexa Fellowship and Tencent AI Lab Rhino-Bird Gift Fund. The conclusions and findings are those of the authors and do not necessarily reflect the views of sponsors.

References

Deepak Agarwal, Bee-Chung Chen, and Bo Pang. 2011. Personalized recommendation of user comments via factor models. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*.

Ali Ahmadvand, Ingyu Choi, Harshita Sahijwani, Justus Schmidt, Mingyang Sun, Sergey Volokhin, Zihao Wang, and Eugene Agichtein. 2018. Emory IrisBot: An open-domain conversational bot for personalized information access. In *Proc. Alexa Prize 2018*.

James F. Allen and C. Raymond Perrault. 1980. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178.

John L. Austin. 1975. *How To Do Things with Words*, 2nd edition. Harvard University Press, Cambridge, MA.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Machine Learning Research*, 3:993–1022.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Constantinos Boulis and Mari Ostendorf. 2005. A quantitative analysis of lexical differences between genders in telephone conversations. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 435–442.

Kevin K. Bowden, Jiaqi Wu, Wen Cui, Juraj Juraska, Vrindavan Harrison, Brian Schwarzmann, Nick Santer, and Marilyn Walker. 2018. SlugBot: Developing a computational model and framework of a novel dialogue genre. In *Proc. Alexa Prize 2018*.

Sandra Carberry. 1983. Tracking user goals in an information-seeking environment. In *Proc. AAAI Conf. Artificial Intelligence*, pages 59–63.

Hao Cheng, Hao Fang, and Mari Ostendorf. 2017. A factored neural network model for characterizing online discussions in vector space. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2296–2306.

David N. Chin. 1986. User modeling in UC, the UNIX consultant. In *Proc. Computer Human Interactions (CHI)*, pages 24–28.

Christopher Cieri, David Graff, Owen Kimball, Dave Miller, and Kevin Walker. 2004. Fisher english training speech part 1 transcripts LDC2004T19. Web Download.

Jennifer Coates, editor. 1998. *Language and gender: a reader*. Wiley-Blackwell.

Jacob Cohen. 1988. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 3079–3087.

David DeVault, Ron Artstein, Grace Benn, Teresa Dey, Ed Fast, Alesia Gainer, Kallirroi Georgila, Jon Gratch, Arno Hartholt, Margaux Lhommet, et al. 2014. SimSensei kiosk: A virtual human interviewer for healthcare decision support. In *Proc. Int. Conf. Autonomous Agents and Multi-agent Systems*, pages 1061–1068.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*.

Hao Fang. 2019. *Building a User-Centric and Content-Driven Socialbot*. Ph.D. thesis, University of Washington.

Hao Fang, Hao Cheng, Elizabeth Clark, Ariel Holtzman, Maarten Sap, Mari Ostendorf, Yejin Choi, and Noah Smith. 2017. Sounding Board – University of Washington's Alexa Prize submission. In *Proc. Alexa Prize*.

Hao Fang, Hao Cheng, Maarten Sap, Elizabeth Clark, Ariel Holtzman, Yejin Choi, Noah Smith, and Mari Ostendorf. 2018. Sounding Board – a user-centric and content-driven social chatbot. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL) (System Demonstrations)*.

Pascale Fung, Anik Dey, Farhad Bin Siddique, Ruixi Lin, Yang Yang, Yan Wan, and Ricky Ho Yin Chan. 2016. Zara the supergirl: An empathetic personality recognition system. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL) (System Demonstrations)*.

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

Dirk Hovy and Anders Søgaard. 2015. Tagging performance correlates with author age. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 483–488.

Eduard Hovy. 1987. Generating natural language under pragmatic constraints. *Journal of Pragmatics*, 11:689–710.

Aaron Jaech and Mari Ostendorf. 2018. Personalized language model for query auto-completion. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 700–705.

Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse relation language models. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*, pages 332–342.

Anders Johannsen, Dirk Hovy, and Anders Søgaard. 2015. Cross-lingual syntactic variation over age and gender. In *Proc. Conf. Computational Natural Language Learning (CoNLL)*, pages 103–112.

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.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learning Representations (ICLR)*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 3294–3302.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*, pages 515–520.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. 2016. A persona-based neural conversation model. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 994–1003.

Diane J. Litman. 1986. Linguistic coherence: a plan-based alternative. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 215–223.

Yang Liu, Kun Han, Zhao Tan, and Yun Lei. 2017. Using context information for dialog act classification in dnn framework. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2170–2178.

Veronica Lynn, Youngseo Son, Vivek Kulkarni, Niranjan Balasubramanian, and H. Andrew Schwartz. 2017. Human centered nlp with user-factor adaptation. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1146–1155.

François Mairesse and Marilyn Walker. 2006. Automatic recognition of personality in conversation. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*, pages 85–88.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 6294–6305.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 3111–3119.

Shachar Mirkin, Scott Nowson, Caroline Brun, and Julien Perez. 2015. Motivating personality-aware machine translation. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1102–1108.

Johanna D. Moore and Cecile Paris. 1992. Exploiting user feedback to compensate for the unreliability of user models. *User Modeling and User-Adapted Interaction*, 2:287–330.

Cécile L. Paris. 1987. *The Use of Explicit User Models in a Generation System for Tailoring Answers to the User's Level of Expertise*. Ph.D. thesis, Columbia University.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*, pages 2227–2237.

Daniel Preoţiuc-Pietro, Wei Xu, and Lyle Ungar. 2016. Discovering user attribute stylistic differences via paraphrasing. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3030–3037.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. 2017. Conversational AI: The science behind the alexa prize. In *Proc. Alexa Prize 2017*.

Abhilasha Ravichander and Alan Black. 2018. An empirical study of self-disclosure in spoken dialogue systems. In *Proc. SIGdial Meeting Discourse and Dialogue*, pages 253–263.

Elaine Rich. 1979. User modeling via stereotypes. *Cognitive Science*, 3:329–354.

John R. Searle. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

Milad Shokouhi. 2013. Learning to personalize query auto-completion. In *SIGIR*, pages 103–112. ACM.

Derek Sleeman. 1985. UMFE: A user modelling front-end subsystem. *Int. J. Man-Machine Studies*, 23:71–88.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Quan Hung Tran, Gholamreza Haffari, and Ingrid Zukerman. 2017a. A generative attentional neural network model for dialogue act classification. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 524–529.

Quan Hung Tran, Ingrid Zukerman, and Gholamreza Haffari. 2017b. A hierarchical neural model for learning sequences of dialogue acts. In *Proc. European Chapter Assoc. for Computational Linguistics (EACL)*, pages 428–437.

Quan Hung Tran, Ingrid Zukerman, and Gholamreza Haffari. 2017c. Preserving distributional information in dialogue act classification. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2151–2156.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 5998–6008.

Svitlana Volkova, Theresa Wilson, and David Yarowsky. 2013. Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1815–1827.

Ingrid Zukerman and Diane Litman. 2001. Natural language processing and user modeling. *User Modeling and User-Adapted Interaction*, 11:129–158.

A Examples for Mode Analysis

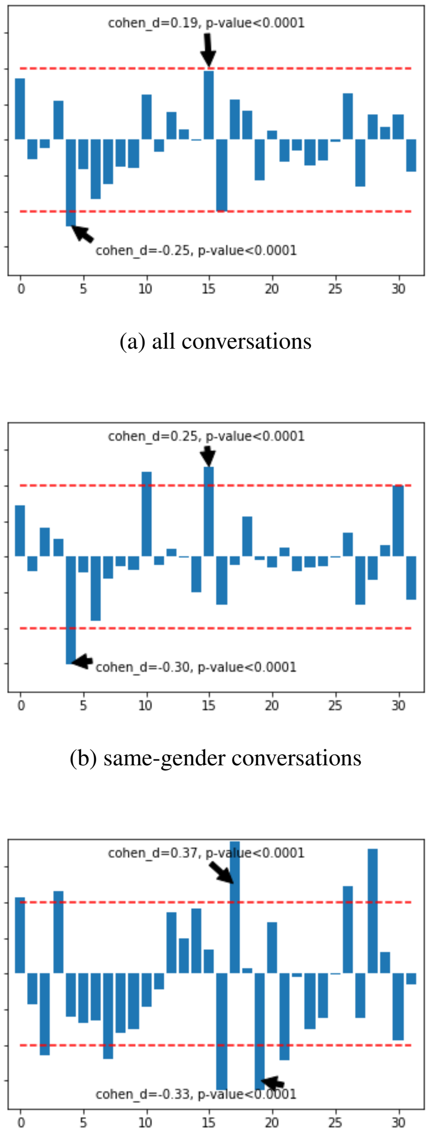
For each mode, we list top associated user utterances in Table 6 and Table 7 for the user topic decision corpus and SwDA corpus, respectively.

For modes learned in the user topic decision corpus, mode 4 seems to include positive reactions, while mode 2 involves slightly negative reactions. Modes 0 and 6 are mostly yes/no answers. Utterances associated with mode 3 are mostly conversation ending. Modes 9, 14, and 10 are mostly set topic commands, differing in style. Mode 10 is associated with complete requests (e.g., "*let's/can we talk about cats*)," while mode 9 and Mode 14 involve short topic phrases (e.g., "*holidays*"). Modes 8 and 11 capture talkative users, whereas modes 1 and 7 capture relatively terse users.

For latent modes learned in the SwDA corpus, there are several modes corresponding to coarse-grained dialog acts, such as statements (modes 2, 4, 6, 16, 19), questions (modes 8, 9), agreement (modes 12, 20), backchannel (modes 0, 28), and conversation-closing (mode 13). Among the statement modes, there are two distinct groups, one (modes 4, 6, 16, 19) containing multiple filled pauses, such as *uh, you know, well*, and the other one (mode 2) with *because*-clauses. The fact that coarse-grained dialog act information is partly encoded in the modes may be helping with recognizing the dialog act.

B Speaker Gender Analysis

We use the speaker gender information from the SwDA data and analyze whether latent modes unsupervisedly learned in the dynamic speaker model could pick up some gender language variations. First, we gather the latent mode association scores for each of the 32 modes for all utterances as computed in (1). Then we carry out the group mean tests for individual modes to test the associate score distributions of male vs female utterances. The Cohen-d score is used to measure the strength of the difference (Cohen, 1988). We also compute the p-value using the Mann-Whitney U test. Previous work has observed larger gender language differences when the two speakers have the same gender (Boulis and Ostendorf, 2005). Thus, we carry out the group mean tests on the following three sets: 1) all conversations, 2) conversations involving only males or females, and 3) conversations involving both genders. The Cohen-d scores for overall, same-gender and cross-gender conversations are shown in Fig. 3. For each set, we identify the most female-like mode (with the most positive Cohen-d score) and the most male-like mode (with the most negative Cohen-d score). For female-like modes, modes 15 and 17 are identified in this way, whereas modes 4 and 19 are identified for male-like modes. By examining representative patterns in modes 15 and 17, they are mostly backchannel, acknowledgement, or agreement. For modes 4 and 19, filled pauses are prevalent.



(a) all conversations

(b) same-gender conversations

(c) cross-gender conversations

Figure 3: Cohen-d scores for gender group tests. The x-axis is the mode index. The y-axis is the Cohen-d score, with a larger magnitude suggesting a large effect size, and a positive value for a more female-like mode. The red dash lines indicate the ±0.20 threshold.

Mode-0	• no no no no no no go back to my alexa . . . • no no no no let's stop talking now goodbye . . . • no let's chat let's chat about donald trump . . .
Mode-1	• gotcha • hiya • possibly
Mode-2	• serious • are you serious • that is a paradox
Mode-3	• alexa resume pandora • alexa connect bluetooth • no bye bye alexa
Mode-4	• that is fascinating • whoa • that that's cool
Mode-5	• i did not that's not surprising • i did not i did not knew that • unfortunately
Mode-6	• somewhat • yes yes yes yes yes • yes i did it was on the news
Mode-7	• mhm • ok • fascinating
Mode-8	• yes it was very much was i saw it i i was there i choose to the dark side did you choose that via uh right . . . • the online selanne jungle the mighty jungle the line the jungle in the jungle the mighty jungle the mighty jungle . . . • no if your life was narrated by someone and the choice was either • i was curious if you 'd rather have your life narrated by regis philbin or by morgan freeman • did you know the answer rogers because like a better go bike and probably i just do n't know it was just a long time ago • i thought bill murray was very very funny
Mode-9	• meow • award shows • celebrity
Mode-10	• no let's talk about butterflies • no let's talk about snakes • can we talk about kardashians
Mode-11	• is king kong real or is he bake but is he awesome or . . . • that is so true the concept of pencils are really stupid and should i even exist imagine if we have pencil do we wanna be able to write on paper so that makes you stupid • is this randomly talking to this is the dawning alligators okay so did we get bored i don't know you somehow or . . .
Mode-12	• do you know alexa how do you how do you know all the stuff you're an a. i. • what what alexa what how do you talk about • alexa do you know alexa do you know a joke today • alexa do you tell me what you know about the new vision nuclear plant
Mode-13	• ten million • thirty percent • what's p. r.
Mode-14	• dog • dogs • tv
Mode-15	• now • not now

Table 6: User utterances in socialbot conversations that have top association scores for individual latent modes.

Statements	Mode-2	• cause i know there ’s one not too far from from me here in dallas • because they really had no idea NONVERBAL what was involved once i got home • because like i said i worked with a lot of those • because he left home at five thirty in the morning • and then she would like to turn in half of the parents that drop their kids off because of the condition the kids are in you know
	Mode-4	• uh some more in interest type topics in in other countries • uh the uh the credit union has got a deal now where you decide what you want • well it would be lower middle class housing here • uh the only other thing i have noticed though is that uh it seems that there ’s been a lot of or more empha emphasis at least in what we ’ve been dealing with
	Mode-6	• and i know that uh you know it can be freezing cold in the wintertime and hot and uh sticky in the summertime • it ’s uh it ’s uh it ’s uh plywood uh face i guess • but i NONVERBAL i i i think you know the biggest causes even then a lot of times are uh uh like when i was up in boston just all the cars you know just all over the place • and so i i it ’s i think i to me i think uh something that ’s going to help our medical uh arena is for um • you know it ’s like it ’s like a luxury car except that it ’s the dodge aries NONVERBAL you know
	Mode-16	• but uh this last ski trip they took uh she had in contracted chicken pox first • but uh we lived in malaysia for t i in nineteen uh eighty one two three and four • well my uh my sister lives in houston • i i was only twenty five years old or something • it ’s uh uh c n n has been a welcome addition to NONVERBAL the t v scene here in the last uh number of years
	Mode-19	• uh i traded off an eighty two oldsmobile for the eighty nine mazda • because i mean after i figured out i was getting eighty cents an hour i said bag it • uh we have a a mazda nine twenty nine and a ford crown victoria and a little two seater c r x. • and uh you know i i was amazed cause i ’d pick up a local paper and i ’d read about all of these you know really interesting things going on • well a friend of mine at work here said that he tried it with his dog
Backchannel	Mode-0	• yes • yes NONVERBAL
	Mode-15	• see • probably • like
	Mode-17	• uh • um
	Mode-18	• oh oh yeah • oh well • oh okay
	Mode-28	• uh huh NONVERBAL • uh huh NONVERBAL NONVERBAL • uh huh ery faint
Agreement	Mode-12	• exactly
	Mode-20	• yep ause • definitely • absolutely • i agree
Quesetion	Mode-8	• are you and your roommate a similar size • did you do the diagnosis or was it just an assumption that that ’s probably the part that failed • or do you have powered you know a • NONVERBAL what kind of a car do you have now • did they know that all along
	Mode-9	• so what do you think about uh what do you think about what you see on t v about them like in the news or on the ads • what do you think about what do you think about the the lower grades you know k through seven • so uh what do you think about our involvement in the middle east • you are talking about p o w s or missing in actions
Conversation-closing	Mode-13	• bye • bye bye • appreciation talking to you

Table 7: Utterances for each mode in SwDA dataset.