

©Copyright 2019

Hao Cheng

Unsupervised Text Representation Learning with Interactive Language

Hao Cheng

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Mari Ostendorf, Chair

Hannaneh Hajishirzi

Luke Zettlemoyer

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Unsupervised Text Representation Learning with Interactive Language

Hao Cheng

Chair of the Supervisory Committee:

Prof. Mari Ostendorf

Department of Electrical and Computer Engineering

Distributed text representations learned through unsupervised learning have recently shown great success in various language processing tasks. However, most of the existing work focuses solely on learning text representations from written documents with little or limited structured context. Different from written documents, dialogues and multi-party discussions contain structured context information in that they take the form of a sequence of turn-taking responses reflecting the attributes of participants and their corresponding communication goals. This thesis aims to represent interactive language with two types of context, *i.e.* local text context and global mode context. An unsupervised text representation learning framework is developed to capture the structured context in interactive language. Experiments show that capturing such context in a text representation can be useful for various language understanding tasks.

An important focus of the proposed unsupervised text representation learning framework is to discover *latent* discrete factors in language. We formulate the latent factor learning as a conditional generation process by dynamically querying a memory of latent mode vectors for template information that is shared across the data samples. A potential advantage of using the latent mode vectors is that they can make the resulting model more interpretable. Based on qualitative analysis, we find the learned latent factors correspond to speaking style, intent, sentiment and even speaker related attributes, such as gender and personality.

The text representation approach is assessed on four interaction scenarios using four different

tasks: 1) community endorsement prediction in multi-party text-based online discussions, 2) topic decision prediction in human-socialbot spoken dialogues, 3) dialogue act prediction in human-human open-domain spoken dialogues, and 4) dialogue state tracking in human-wizard text-based task-oriented dialogues. The resulting text representation is shown to be effective for all four scenarios, demonstrating the benefit of incorporating interaction context with the unsupervised text representation learning.

TABLE OF CONTENTS

| | Page |
|--|------|
| List of Figures | iii |
| List of Tables | v |
| Chapter 1: Introduction | 1 |
| 1.1 Our Approach and Contributions | 3 |
| 1.2 Dissertation Overview | 5 |
| Chapter 2: Background | 7 |
| 2.1 Neural Text Representations | 7 |
| 2.2 Neural Encoder-decoder with Attention | 9 |
| 2.3 NLP for Interactive Language | 12 |
| 2.4 Limitations of Existing Work | 16 |
| Chapter 3: General Approach | 18 |
| 3.1 Terminology | 18 |
| 3.2 Model Components | 19 |
| Chapter 4: Multi-party Online Discussions | 23 |
| 4.1 Model Description | 25 |
| 4.2 Model Learning | 29 |
| 4.3 Data and Task Details | 30 |
| 4.4 Experiments | 32 |
| 4.5 Qualitative Analysis | 36 |
| 4.6 Summary | 38 |
| Chapter 5: Open-domain Social-chat Dialogues | 40 |
| 5.1 Dynamic Speaker Model | 41 |

| | | |
|--------------|--|----|
| 5.2 | User Topic Decision Prediction | 45 |
| 5.3 | Dialog Act Classification | 55 |
| 5.4 | Summary | 60 |
| Chapter 6: | Task-oriented Dialogues | 63 |
| 6.1 | Background | 63 |
| 6.2 | Dialogue State Tracking Model | 65 |
| 6.3 | Experiment | 68 |
| 6.4 | Summary | 71 |
| Chapter 7: | Conclusion | 72 |
| 7.1 | Summary | 72 |
| 7.2 | Impacts and Implications | 73 |
| 7.3 | Future Directions | 74 |
| Bibliography | | 77 |

LIST OF FIGURES

| Figure Number | Page | |
|---------------|---|----|
| 4.1 | The structure of the full model omitting output layers, illustrating the computation of attention weights for \mathbf{b}_2 and \mathbf{d}_3 in a comment $w_{1:4}$ with its response $r_{1:4}$. Purple circles a_k and a'_j represent scalars computed in (4.2) and (4.8), respectively. \otimes and \oplus are scaling and element-wise addition operators, respectively. Black arrowed lines are connections carrying weight matrices. | 25 |
| 4.2 | Averaged F1 scores of different classifiers. Blue bars show the performance using no comment embeddings. Orange bars show the absolute improvement by using factored comment embeddings. | 30 |
| 4.3 | The quantized karma distribution. | 31 |
| 4.4 | F1 scores of the DeepOR classifier for individual subtasks. Error bars indicate the improvement of using the factored comment embeddings over the classifier using no text features. | 34 |
| 4.5 | The confusion matrices for the DeepOR classifier on <code>Politics</code> . The color of cell on the i -th row and the j -th column indicates the percentage of comments with quantized karma level i that are classified as j , and each row is normalized. | 35 |
| 4.6 | The box plot of strongest association positions for each global mode in <code>Politics</code> | 37 |
| 4.7 | t-SNE visualization of response trigger vectors clustered using k-means. | 38 |
| 5.1 | 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 | 42 |
| 5.2 | 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. | 56 |
| 5.3 | 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. | 59 |

| | | |
|-----|---|----|
| 6.1 | A dialogue state tracking example. In this example, a user is looking for a moderately priced Chinese restaurant in the east side. Turn labels are shown in blue boxes, and dialogue states are shown in green boxes. | 64 |
| 6.2 | Dialogue state tracking using two dynamic speaker models for agent and user, respectively. The dialogue state tracker makes slot value predictions using output from both models. | 65 |

LIST OF TABLES

| Table Number | Page |
|--|------|
| 1.1 Characteristics of the four interaction scenarios studied in this thesis. | 4 |
| 2.1 Three encoder-decoder paradigms. | 10 |
| 2.2 Example for illocutionary and perlocutionary forces between a speaker and a listener. | 10 |
| 4.1 Averaged F1 scores of DeepOR classifiers using different text features. Baseline results do not use any text features. | 33 |
| 4.2 Examples of comments associated with the learned global modes for <code>Politics</code> . . | 39 |
| 5.1 A sample conversation between the user and Sounding Board. There are two user-requested topics: <code>Superman</code> (turn 5), <code>science</code> (turn 11). There are three bot-suggested topics: <code>Henry Cavill</code> (turn 8), <code>movie Superman</code> (turn 10) and <code>car</code> (turn 14). A suspected speech recognition error in the user utterance at turn 3 is shown in []. | 47 |
| 5.2 Data statistics of the topic decision dataset. | 48 |
| 5.3 Test set results (in %) for topic decision predictions using <i>static</i> user embeddings. . | 50 |
| 5.4 Test set results (in %) for topic decision predictions using <i>dynamic</i> user embeddings. *: The improvement of <code>DynamicSpeakerModel</code> over both <code>TopicDecisionLSTM</code> and <code>UtteranceAE + LSTM</code> is statistically significant based on both t-test and McNemar’s test ($p < .001$). | 50 |
| 5.5 A dialog snippet showing topic decision predictions from <code>TopicDecisionLSTM</code> and <code>DynamicSpeakerModel</code> . Topics are shown with underscores. | 52 |
| 5.6 User utterances in socialbot conversations that have top association scores for individual latent modes: modes 0–7. | 53 |
| 5.7 User utterances in socialbot conversations that have top association scores for individual latent modes: modes 8–15. | 54 |
| 5.8 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 ($p < .001$). | 58 |
| 5.9 Utterances for each mode in SwDA dataset: statements. | 61 |

| | | |
|------|--|----|
| 5.10 | Utterances for each mode in SwDA dataset: other dialog acts. | 62 |
| 6.1 | Data statistics of the Single-WoZ dataset and the Multi-WoZ dataset. | 68 |
| 6.2 | Joint goal accuracy on the Single-WoZ test set. | 69 |
| 6.3 | Joint goal accuracy on the Multi-WoZ test set. | 70 |

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Mari Ostendorf. During my PhD studies, I have learned a lot from her insightful advice on improving my research, writings, and public speaking skills. Moreover, she has always been supportive, letting me pursue my interest in different directions.

I would also like to thank my reading committee members, Hannaneh Hajishirzi and Luke Zettlemoyer, for your time, your guidance, and help in improving my thesis. Thanks to Cecilia Aragon for serving the graduate school representative.

Many thanks to Sounding Board team members, Hao Fang, Elizabeth Clark, Ari Holtzman, and Maarten Sap, for all the hard-working days we had together on developing the bot and our great memories of the award-winning moment. Also thanks to both Yejin Choi and Noah A. Smith for advising the team together with Mari.

I have also been fortunate to be part of a vibrant and supportive research lab. Thank you to all TIAL lab members and alumni: Kevin Everson, Ji He, Jingyong Hou, Aaron Jaech, Michael Lee, Yuzong Liu, Roy Lu, Yi Luan, Kevin Lybarger, Alex Marin, Farah Nadeem, Sara Ng, Sining Sun, Trang Tran, Ellen Wu, Sitong Zhou, Victoria Zayats. Thanks to our lab's system administrator, Lee Damon, who has maintained a stable filesystem and network for the lab.

I have had great internships at Microsoft Research and Google AI Language. I would like to thank all my mentors and collaborators: Ming-Wei Chang, Michael Collins, Li Deng, Xiaodong He, Jianfeng Gao, Kenton Lee, Ankur Parikh, Hoifung Poon, Chris Quirk, Kristina Toutanova, Scott Wen-tau Yih.

Finally, I would like to thank the support of my grandparents: Yinsheng Cheng and Yuefeng Zhu. It is their love that makes it possible for me to complete my PhD journey.

Chapter 1

INTRODUCTION

Continuous-space representations of language (a.k.a. embeddings) have had a tremendous impact on natural language processing (NLP). Traditionally, text features, usually categories derived either from surface form or linguistic annotation, are heavily hand-engineered based on specific domain knowledge. This type of text representation also has to deal with the exponential dimension problem when representing higher-order feature dependencies, which leads to data sparsity and further compromises the generalization ability. In contrast, the distributional hypothesis for text embeddings assumes the meaning of words is embodied in the surrounding text, which can be learned from massive text resources. Compared with their hand-engineered counterpart, embeddings learned automatically from large corpora promise better generalization ability.

Representing language in context is key to improving 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. In this thesis, we focus on two types of context, *i.e.* *local text context* and *global mode context*. Here, “local text context” refers to neighboring words or sentences. Generally, for written documents, the local text context is linear (a flattened sequence of words or sentences), although structured context can be derived from local text context by using linguistic annotations, such as word dependencies or discourse. On the contrary, interactive language is inherently structured in that sentences are segmented into turns associated with alternating interlocutors. Different from local text context, we use “global mode context” to refer to characteristics of the interlocutor, social context and/or community identity that impact language use. For example, in online discussions, the community identity highly influences how participants communicate on discussion forums, such as picking topics and making responses. Furthermore, different personality traits can lead to different speaking styles when expressing sim-

ilar dialogue acts that are common in social-chat conversations. In this thesis, we are interested in deriving representations of both local text context and global mode context. Particularly, we focus on representing sentences or short sequences of sentences, such as a spoken turn in a dialogue.

Findings have shown that word embeddings learned in an unsupervised manner through predicting the local text context, *i.e.* the next word or surrounding context words, are very effective at capturing syntactic and semantic information [1–3]. These word embeddings are usually used to initialize a specialized model for some downstream task, and it is often necessary to fine-tune them jointly with the target task-dependent model. This usage of word embeddings has proven to be effective in many NLP tasks including both natural language understanding and natural language generation.

Motivated by the success of neural network word embeddings, researchers have developed neural-based representations for larger units of text, such as sentences or documents [4, 5]. In many NLP tasks such as sentiment analysis and document classification that mainly deal with written documents, context effects on the prediction are dominated by the local sequence structure within a sentence, or sometimes sentence sequences. However, for NLP tasks associated with interactive language, such as spoken dialogue or multi-party online discussion, the assumption of linear context is too limiting. For example, there are multiple participants contributing to the conversation, and language understanding benefits from knowing who said what. In other words, the communication goals and speaker attributes, such as topic preference and speaking style, play an important role in understanding and responding to a sentence or utterance. Therefore, this thesis develops new unsupervised learning techniques that take advantage of structured context in interactive language, which include the structure of discussions or dialogues, and community or user verbal reactions.

Although embedding representations allow effective unsupervised learning from text, existing approaches mainly produce a single global representation for the text unit. Specifically, most existing work represents words [1,3], sentences, or documents [4,5] with a single vector. Typically, the models are trained using a language prediction task within a certain local text context window. For words that can have different senses, the learned global representation usually reflects a mixture

of contexts, *i.e.* different syntactic and semantic roles, according to the contexts in the training data. However, for many language understanding tasks, a more localized (contextualized) representation is likely to be more effective, explicitly using the local context of the word. Inspired by that, a line of work has been developed to learn locally-contextualized word representations, relying on either extra structured knowledge in the form of multiple objectives [6] or the composition of multiple embeddings learned from large training sets with richer context to capture polysemy [7–9]. Compared with single vector word embeddings, contextualized word embeddings, especially more recent ones such as ELMO [8] and BERT [9], have brought significant improvement across numerous of NLP tasks.

Similar to individual words, sentences carry semantic meaning, intent, sentiment, affect, and more. Depending on the discourse/social context, a single sentence can have multiple meanings. For example, the sentence “*I’m good*” can be used to answer the question “*How are you?*” or to express a polite decline of an offer; the specific context is required to resolve the ambiguity. Existing work either focuses on representing intra-sentence information [5] or restricting the inter-sentence representation to mainly linear context [4]. In this thesis, we focus on contextualizing sentence representations by utilizing the structured context in interactive language. Motivated by the multi-faceted nature of interactive language, we introduce a factored sentence representation in which each subvector corresponds to a different salient factor such as topic preference and speaking style. Moreover, those subvectors are grounded on the corresponding context which makes the overall representation contextualized.

1.1 Our Approach and Contributions

In this thesis, we build an unsupervised representation learning framework through the conditional neural language model (NLM) with multiple subvectors, each of which captures a different salient factor of the interactive language. Our view is that factoring the model of language generation and training with multiple objectives associated with structured context in interactive language can lead to unsupervised learning that effectively subdivides the larger vector into more interpretable components. Moreover, factoring the model based on language interaction provides a mechanism

| | Online Discussion | Human-Socialbot Dialogue | Human-Human Social Chat | Task-Oriented Dialogue |
|--------------------------------|----------------------------|---------------------------------|-----------------------------------|---------------------------------------|
| Communication Goal | Information Sharing | | | Task Completion |
| Turn-taking | Asynchronous | Coordinated | | |
| Participants | Multi-Party Human-Human | Human-Computer | Human-Human | Human-Wizard |
| Genre | Written Text | ASR Transcripts & System Log | Spontaneous Speech Transcripts | Wizard-of-Oz Text Conversation Log |
| Model Prediction Target | Community Endorsement | Topic Decision | Dialogue Act | Dialogue State |

Table 1.1: Characteristics of the four interaction scenarios studied in this thesis.

to explore language processing problems that have been less studied in NLP. Specifically, we are interested in tasks that require capturing the language interaction context. Based on this, we select four representative interaction scenarios for evaluation: online discussion, human-socialbot dialogue, human-human social chat, and task-oriented dialogue. Table 1.1 compares these scenarios in terms of five main aspects: communication goal, turn-taking, participants, genre and model prediction target.

Besides the factored model, another important focus of the proposed unsupervised learning framework is to discover discrete *latent* factors in the language. Unsupervised learning strategies with discrete latent variables have been explored previously mainly to characterize the topical information which has been shown to be an important feature for downstream tasks, such as text classification or model adaptation [10, 11]. In this thesis, we formulate the latent variable learning as a conditional generation process by dynamically querying the memory for template information that is shared across the data samples. Specifically, we model latent factors that are pervasive in interactive languages, such as speaking style, intent, sentiment and so on, which would be jointly trained with a language generation objective.

To sum up, this thesis makes the following contributions:

1. We introduce a new unsupervised factored representation learning framework for interactive

language, where the resulting representations are aware of both local text context and global mode context.

2. As shown in Table 1.1, we apply the proposed text representation to four representative interaction scenarios: online discussions, human-socialbot dialogues, human-human social chat dialogues and task-oriented dialogues. Each scenario is evaluated on its corresponding task, *i.e.* community endorsement prediction, topic decision prediction, dialogue act prediction and dialogue state tracking. Experiments show that the proposed text representation can benefit all four tasks associated with different types of interactive language.
3. Lastly, we extend the conditional sequence model with joint *discrete* latent information inference. The learned latent factors can be potentially associated with certain interpretable salient aspects in different types of data. Through qualitative analysis, the model is found to be capable of capturing both community-level or user characteristics which results in improvement over multiple predictive tasks under different interactive language scenarios.

1.2 Dissertation Overview

In Chapter 2, we first provide a general overview of related recent work on neural text representation and neural encoder-decoder attention model. Then, interactive language processing tasks are reviewed in terms of types of interactive language, applications that use interactive language, and technologies that support those applications. We also discuss limitations of existing work on unsupervised text representations for interactive language.

Chapter 3 introduces the terminology that is used for describing the interactive language studied in this thesis. After that, we formally present the two building blocks that are used to design unsupervised text representation models for interactive language, *i.e.* an encoder-predictor model and a latent dynamic context model. In addition to a unifying view of multiple sequential models, the encoder-predictor model is leveraged to derive factored representations where multiple sub-vectors are used to model different types of context. The latent dynamic context model provides a mechanism for joint learning of global latent mode context but also enables the possible interpretation of what is learned by each mode.

Based on the general approach, a factored neural model is developed in Chapter 4 to jointly learn (in an unsupervised fashion) aspects that impact comment reception in multi-party discussions from a large collection of Reddit comments. The multi-factored comment embedding is evaluated on the task of predicting the comment endorsement for three online communities differing in topic trends and writing style. Moreover, the global modes and response trigger subvectors are analyzed to show what kind of language factors are captured in the resulting embeddings.

In Chapter 5, we focus on extending our unsupervised text representation learning approach to capture speaker-related language factors that can be used as context in dialogue-centered language understanding tasks. We evaluate the dynamic speaker model on two tasks for open-domain social chat dialogues: predicting user topic decisions in human-socialbot dialogues, and classifying dialogue acts in human-human dialogues. In addition, analyses are performed to align the learned characteristics of a speaker with the human-interpretable factors.

In Chapter 6, we apply the dynamic speaker model to task-oriented dialogues. Specifically, we develop a neural dialogue state tracking model based on the dynamic speaker model, which can be pretrained on task-oriented dialogues to capture features useful for dialogue state prediction. In order to evaluate the proposed model, two recent dialogue state tracking datasets collected in the Wizard-of-Oz fashion are studied.

Finally, in Chapter 7, we conclude the thesis by summarizing the approach, models developed, and experimental findings. Since this thesis represents early work on developing unsupervised contextualized representations for interactive language, we also discuss the impacts and implications of the contribution. Lastly, we outline some future directions for extending the factored text representation learning and other potential applications.

Chapter 2

BACKGROUND

While many existing approaches have been developed to learn neural text representations from written documents, the focus of this thesis is on contextualized representations for sentences in interactive language, *i.e.* online discussion and dialogues. In this chapter, we first review existing neural text representations in §2.1. Given our model is built upon the neural encoder-decoder with attention mechanism, we then review this architecture in §2.2. In §2.3 we discuss different types of interaction scenarios and review NLP applications and technologies for interactive language. Finally, we conclude this chapter in §2.4 with discussion of the limitations of existing work.

2.1 Neural Text Representations

Neural networks have been used in a variety of ways to learn continuous word embeddings in an unsupervised manner. Feedforward neural networks and recurrent neural networks are two general examples, where the continuous word embedding can be derived from the connection between the one-hot input layer and the first hidden layer, or that between the last hidden layer and the softmax output layer. Both the feedforward NLM [12] and the log-bilinear (LBL) LM [13] use the feedforward structure, but they differ in the choice of hidden layer activation function and the parameter-sharing mechanism. The continuous bag-of-word (CBOW) model and the skip-gram model are variants of the feedforward NLM, which explore both left and right context words simultaneously [1]. Similarly, the vLBL model and the ivLBL model are such variants of the LBL LM [14]. In [15], the continuous word embeddings are learned from a model analogous to the skip-gram model that uses the syntactic context words derived from dependency parsing. Unlike the feedforward neural networks in which history words in a context window are projected in parallel to a hidden layer, the recurrent NLM obtains a hidden layer by recursively performing

projection and non-linear transformation [16]. The continuous word embeddings derived from the connection between the input layer and the hidden layer of the recurrent NLM are shown to be capable of capturing syntactic and semantic regularities characterized by relation-specific vector offsets [2]. Different from early work where a single vector is used for representing a word globally, recent work focuses on developing contextualized word embeddings by representing the word together with the corresponding text context. Specifically, a context window where the word appears is encoded using either LSTM or Transformer architecture [17] to perform language prediction tasks, such as machine translation [7], bi-directional language modeling [8], masked language modeling [9], *etc.*

Inspired by the work on continuous word embeddings, people have begun to explore continuous embeddings for a sequence of words, usually called continuous sentence embeddings. Some work directly learns such embeddings based on aggregation of word embeddings. By treating the sentence as a bag of words or a bag of n-grams, the deep structured similarity model [18] and the convolutional latent semantic model [19] are proposed for information retrieval. In [5], the authors extend the CBOW and the skip-gram models to paragraph vector models, which learn the paragraph vectors with distributed memory and distributed bag-of-words, respectively. For the paragraph vector with a distributed memory model, a vector representing the paragraph and the word embeddings within the paragraph context are concatenated or averaged in order to predict the current word. The paragraph vector with a distributed bag-of-words model uses a vector representing the paragraph to predict words in a context window. During inference, the word embeddings and network parameters are fixed, and paragraph vectors are obtained by gradient descent. Other studies use different composition methods to obtain sentence embeddings from word embeddings. In [4], the authors leverage the embeddings of surrounding sentences to improve the sentence embeddings, analogous to learning the word embeddings from context words in the skip-gram model. A time-delay neural network (also known as convolutional neural network) with a max-pooling-over-time operator is used to compose the sentence embeddings in [20], by using an auxiliary unsupervised language modeling task in addition to the target classification tasks. In addition, there is a large body of work focusing on using different neural architectures to build sentence representations for super-

vised language processing tasks. In [21], a general max pooling operator, named dynamic k -Max pooling, is used with the convolutional neural network to learn the sentence embeddings. The recurrent neural networks compose a sentence embedding recursively from left to right, e.g., in [22], whereas the recursive neural networks compose phrase/sentence embeddings recursively according to a tree structure [23, 24]. In [25], a recursive convolutional neural network is proposed, which composes the sentence embedding by recursively applying the weights of a binary convolutional neural network to the input sequence. Advanced hidden units are usually used for these recurrent and recursive neural networks, e.g., the long short-term memory (LSTM) unit and its variants [26], and the gated recurrent unit (GRU) [27]. Furthermore, while intra-sentence structure information has been explored for learning continuous sentence embeddings, e.g., in [24, 28], work leveraging inter-sentence structure is in very early stages.

Although various approaches have been developed for representing text in written documents, to our knowledge, there is no work developed explicitly for interactive language. In this thesis, we develop an unsupervised representation learning framework, which explicitly reflects speaker/writer turn changes in language interactions.

2.2 *Neural Encoder-decoder with Attention*

The neural encoder-decoder model [29] has been successfully applied to many NLP tasks, such as machine translation [29, 30], summarization [31], syntactic parsing [32], image captioning [33], and chitchat conversation generation [34]. Since this encoder-decoder framework is a key component of the proposed modeling framework, we review the existing encoder-decoder paradigms here.

As shown in Table 2.1, there are three types of encoder-decoder models, which we refer to as:

- **auto-encoder model**: the main goal is to derive lower-dimensional latent representations by reconstructing the corresponding input in an unsupervised fashion,
- **translation model**: a reconstruction decoder is used to generate the target output which generally relates to the given input in another form (different modality or language), thus requiring pairing of inputs and outputs,

| Input | Output | Model | Supervision | Decoder Model |
|-------|--------|-------------------|-------------|---------------------------|
| X | X | Autoencoder | No | Reconstruction |
| X | X' | Translation Model | Yes | Reconstruction |
| X | Y | Interactive Model | No | Directional Trigger Model |

Table 2.1: Three encoder-decoder paradigms.

| Utterance | Illocutionary Force | Perlocutionary Force |
|------------------------|---------------------|---------------------------------|
| Can you pass the salt? | Request | (Response): Sure. Here you are. |
| I want some salt. | Request | (Response): It's on the table. |
| Salt! | Request | (Response): Get it yourself! |

Table 2.2: Example for illocutionary and perlocutionary forces between a speaker and a listener.

- **interactive model:** conditioned on the input (cause), the decoder aims to predict the paired output (effect), where the input-output pair is in a temporal order with potential cause and effect relationship.

One key feature that differentiates the auto-encoder and translation model is that the goal of the decoder is to reconstruct the output sentences based on the information encoded in another parameter space, such as another language with the same meaning or salient concepts in the same language. Because of that, the reversed input-output model would typically still fall into the same category. They also have some different characteristics. In contrast, an auto-encoder does not require any additional supervision. In other words, it is purely self-reconstruction and a fully unsupervised model. Typically, this would result in learning representations with lower dimension or codes with latent information. The translation model usually requires explicit supervision by pairing up meaningful input-output sequences. For example, typical translation models, such as neural machine translation, image captioning and summarization, require human annotations, *i.e.* associating a target sentence to a source sentence, or generating a caption/summary for a given image/article.

The major difference between the interactive model and the aforementioned models is that input-output for the interactive model is directional, *i.e.* the order of the input-output pair cannot

be reversed. In other words, the output is usually the consequence triggered by the input. For this reason, we will refer to this type of model as “encoder-predictor” rather than “encoder-decoder”. Some recent work on the interactive model includes neural conversational models [34, 35] and neural-based spoken dialogue systems [36, 37]. If we assume a comment and response pair is taken from a conversation, for the interactive model, the directional generation decoder module, composes the reply based on the topic, intent and affect captured as conditional information and the user’s own goals. Specifically, take the speech act theory [38] to illustrate the difference between the reconstruction model and the directional trigger model. The illocution, including the semantics and intent, can be typically captured by a reconstruction model, where the politeness might be modified by a translation model. As shown in Table 2.2, although the surface forms are different, *i.e.* question, declarative or imperative, the illocutionary force of the speaker, mainly the underlying intent, is always the same, *i.e.* making a request. However, the feeling that the utterance might produce in the listener would not be the same, *e.g.* positive towards politeness and negative towards rudeness. This kind of directional triggering effect is very different and can be important in representing interactive language. Both are also dependent on the social context. For example, with different social relationships, the same utterance would receive different reactions.

The neural encoder-decoder model with attention, also known as memory networks [39], have been widely used in many tasks, *e.g.*, machine translation [29, 30], question answering [40, 41], syntactic parsing [32], as well as graph-based dependency parsing [42–45]. In [29], the decoder utilizes a weighted sum of vectors from all time steps of the encoder. The attention weights are dynamically changed during decoding, allowing the decoder to pay attention to different parts of the source text at each time step. In [39], the authors describe the attention mechanism in a more general memory network framework, where the memory components act as the encoder in the sequence-to-sequence model, and it consists of a set of input memory vectors and a set of output memory vectors. With a query vector, the input memory vectors are used to compute the attention weights for individual output memory vectors, and the output memory vectors are used for final prediction.

2.3 NLP for Interactive Language

Social interaction is a primary function of language. There exist two popular views of language interaction: one views language as primarily communicative in function as the “conduit metaphor”, and the other views language as medium of organized social activity. The conduit metaphor is based on the that, through speech, a speaker conveys information by realizing it in words and sending them along a communicative channel. Other speakers then receive the message and extract thoughts and feelings from them. On the other hand, in terms of speech act theory [38], language is a site of social activity, *i.e.* utterances are performative rather than referential. Specifically, the locutions through which speakers provide information about their thoughts or feelings occur as part of some context of acting and are illocutionary. Therefore, the “same” utterance can perform a variety of different speech acts as shown by the “*I’m good*” example in Chapter 1. The modeling framework developed in this thesis is mainly motivated by the second view of language, *i.e.* speech act theory [38]. In this section, we first discuss different types of interaction scenarios in §2.3.1. Then in §2.3.2, we review NLP applications and technologies for interactive language.

2.3.1 Social Interaction Scenarios

One way to categorize social interaction scenarios is based on whether the *written text* or *spontaneous speech* is used for communication. Written text interaction scenarios typically take place on online social media, such as Twitter, Reddit, Quora, *etc.* Similar to written documents, the authors can read the entire discussion history or even external documents to tailor their writings. However, different from documents, the written text is used to fit their communication goals for immediate reactions. Typically, the text content can be either a directed response or a broadcast message within a certain community, which highly influences the topic preference and word usage of participating members. In addition to that, the communication is open to all interested participants, and thus, this type of interaction is often multi-party. On the other hand, most of our daily human-human dialogues are carried out using spontaneous speech, such as multi-party or two-party meetings, two-party customer service calls, *etc.* One additional notable spontaneous speech-based interac-

tion scenario takes place between a human user and a dialogue system powered by conversational AI, such as Amazon Alexa, Apple Siri, Google Assistant, *etc.* For two-party dialogues, the speakers often use directed responses. Different from the written text counterpart, spoken interactive language tends to be full of filled pauses (*um, uh*)/repetitions, incomplete sentences, corrections and interruptions. Moreover, some types of vocabulary are used only or mainly in speech, including slang expressions and tags such as “y’know”, “like”, *etc.* Moreover, different from written documents, automatic speech recognition (ASR) transcripts inevitably contain recognition errors and are comprised of unsegmented utterances.

2.3.2 *NLP Applications and Technologies for Interactive Language*

Many NLP applications relies on interactive language, such as social network analysis, automatic summarization, conversational system, *etc.* Social network analysis focuses on gathering and analyzing data from social media using analytic tools for decision making. In addition to leveraging the social network structure information, recent work uses language analytic tools including sentiment analysis [46], popularity prediction [47, 48], virality detection [49] for tracking trends and identifying important components, such as persuasive strategies [50–52] and community endorsement [53, 54]. Automatic summarization aims at shortening formal documents or informal text, such as online discussions or meetings, in order to create a summary preserving the major points from the original text to increase people’s productivity. Traditionally, both extractive [55] and abstractive summarization [56] focus more on formal written document. Recently, there is an increasing amount of work developing summarization techniques for informal text, such as online user-generated content [57] and spoken meeting transcripts [58]. Lastly, there has been a long line of efforts in developing conversational systems. In the literature, the two main types of conversational systems are task-oriented and non-task-oriented systems. Task-oriented systems primarily focuses on accomplishing user specific goals that range from single and well-defined tasks (*e.g.* hotel or flight booking, restaurant reservation) to more complex tasks involving reasoning and context-switching (*e.g.* trip planning, contract negotiation). In contrast, non-task-oriented systems usually engage users in interactions that do not necessarily involve a specific task. Such systems

(a.k.a. chatbots) have been developed for entertainment, companionship and education purpose. More recently, the Alexa Prize socialbot is a novel application of conversational AI [59]. Significantly different from task-oriented systems and chatbots, Alexa Prize socialbots are developed with the goal of discussing popular topics and recent events.

Community Reaction Prediction: Massive user-generated content on social media such as Twitter and Reddit has drawn interest in predicting community reactions in the form of virality [49], popularity [47,48,60,61], community endorsement [53,54], persuasive impact [50–52], etc. Many of these studies have analyzed content-agnostic factors such as submission timing and author social status, as well as language factors that underlie the composition of the textual content, e.g., the topic and idiosyncrasies of the community. In particular, there is an increasing amount of work on online discussion forums such as Reddit that exploits the conversational and community-centric nature of the user-generated content [50–54,61], which contrasts with Twitter content where the author’s social status seems to play a much larger role in popularity. In this thesis, we focus on Reddit, using the karma score¹ as a readily available measure of community endorsement.

A number of studies have examined multiple factors that influence community endorsement in *Reddit* discussions using different tasks. Althoff et al. estimate the likelihood of successful pizza requests in the `RandomActsOfPizza` subreddit [51]. Jaech et al. propose a comment ranking task controlling the topic and timing factors in order to study how different language factors influence the comment karma scores [53]. Both [50] and [52] work with persuasive arguments in the `ChangeMyView` subreddit and study factors that may help winning arguments or identifying persuasive comments. The classification task of quantized karma is used in [54], however, without characterizing latent language factors or exploiting the interaction between a comment and its replies.

User Modeling: As reviewed by [62], user modeling for conversational systems has a long history. The research can be tracked back to the GRUNDY system [63] which categorizes users in terms of hand-crafted sets of user properties for book recommendation. Other systems have focused on

¹The karma score of a comment is computed as the difference between up-votes and down-votes.

different aspects of users, e.g., the expertise level of the user in a specific domain [64–67], the user’s intent and plan [68–71], and the user’s personality [72–75]. 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 [75], or a conditional random field sequence model with hand-crafted user and context features [76]. Modeling speakers with continuous embeddings for neural conversation models is studied in [35], 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 variation has been analyzed in previous work and incorporated in NLP models. In [77], the authors discover that the user attribute leads to different stylish phrase choices. In addition, based on universal dependencies, Johannsen et al. [78] find that there exists syntactic variation among demographic groups across several languages. Motivated by those findings, speaker demographics are used to improve both low-level tasks such as part-of-speech tagging [79] and high-level applications such as sentiment analysis [46] and machine translation [80]. More recently, in [81], a continuous adaptation method is introduced 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.

Dialogue State Tracking: Dialogue state tracking in task-oriented dialogue systems has been proposed as a part of dialogue management and aims to estimate the belief of the dialogue system on the state of a conversation given the entire previous conversation context [82]. Traditionally, the dialogue state tracking challenges (DSTC) [83] focus on developing dialogue state trackers for datasets curated from human-computer systems, while more recent developments focus more on clean datasets collected in the Wizard-of-Oz fashion [84]. Many dialogue state tracking models utilize Spoken Language Understanding (SLU) outputs [85], which can suffer from the accumulated errors from the SLU. Different from traditional approaches relying on delexicalization of

slots and values in the utterance with generic tags [86], recent end-to-end approaches that directly estimate the states from natural language input based on neural networks have achieved remarkable success [84, 87, 88] on conversation logs either collected in the Wizard-of-Woz fashion or automatically transcribed from spoken dialogues.

2.4 *Limitations of Existing Work*

Although various approaches have been developed for representing text in monologues, to our knowledge, there is no work developed explicitly for interactive language. In this thesis, we take the first step to develop an unsupervised representation learning framework for interactive language. Motivated by the recent success of unsupervised contextualized word embeddings, our approach is to use unsupervised learning with a neural model to contextualize a sentence representation based on the discussion tree or dialogue history. By utilizing the context structure in interactive language, the resulting representation can be more useful for downstream predictive tasks, such as content recommendation, dialogue act prediction and dialogue state tracking.

While some of the prior work on Reddit investigates specific linguistic phenomena (e.g. politeness, topic relevance, community style matching) using feature engineering to understand their role in predicting community reactions as discussed in §2.3.2, we develop the first unsupervised text representation with the awareness of the discussion structure. Furthermore, we structure the text embedding model so as to provide some interpretability of the results when used in comment endorsement prediction. The resulting model characterizes the interdependence of a comment on its global context and subsequent responses that is characteristic of multi-party discussions. Specifically, in Chapter 4, we present a factored neural model with separate mechanisms for representing global context, comment content and response generation. By factoring the model, we show that unsupervised learning picks up different components of interactive language in the resulting embeddings, which leads to improved prediction of community reactions evaluated on three representative communities from Reddit.

Lastly, accounting for author/speaker variations has been shown to be useful in many NLP tasks. 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, speaking style and communication goal. Our approach is to use unsupervised learning with a neural model of a speaker's dialog history. We design our unsupervised model so that model analysis can be performed to find out what the model learns about speaking style. Further, the model is structured to allow a dynamic update of the speaker state representation at each turn in a dialogue, in order to capture changes over time and improve the speaker representation with more interactions. The speaker embeddings can be used as context in conversational language understanding tasks. In Chapter 5, we first apply the speaker model to open-domain social chat dialogues for dialog policy prediction in human-computer dialogues and dialog act prediction in human-human dialogues. Furthermore, in Chapter 6, we show that the speaker embedding can also be used for dialogue state tracking in task-oriented dialogues.

Chapter 3

GENERAL APPROACH

In this thesis, we aim at learning neural representations for text with interactive structure under the unsupervised learning setting. We factor the representation with regard to two types of context in interactive languages: 1) local text context consisting of preceding words and sentences, and 2) global mode context associated with a user or a community, which may capture factors such as topic preference, interaction patterns (*e.g.* dialogue act sequences), or idiosyncrasies of language use. In Section 3.1, we first set up the terminology that will be used for describing the interactive language. Next, we discuss the common neural network building blocks in Section 3.2 which can be adapted based on the context structure of interest.

3.1 Terminology

Since the proposed framework can be applied for both multi-party discussions and two-party dialogues, it is useful to establish common terminology. The term **user** will be used to refer to both authors contributing to a discussion forum or human speakers conversing with a bot or another human speaker. The term **sentence** will be used to refer to both written sentences and spoken utterances.

For multi-party online discussions, the term **comment-response pair** will be used to denote a sequential pair of a comment and the corresponding response. Both comment and response will be in the form of a word sequence including multiple sentences. Here, the **global mode context** will be used to characterize the discussion forum community, and the **local text context** will denote both the preceding words in the comment and the corresponding responses to the current comment.

For two-party dialogues, the two users take turns contributing to the conversation. Here, the

term **turn** will be used to indicate a sequence of sentences from one side between turns taken by the other party. For human-computer or Wizard-of-Oz dialogues, we will denote the computer as **agent**. Here, the **global mode context** will characterize a user, and the **local text context** is the dialogue history of interest, specifically the preceding turns from the user side.

3.2 Model Components

In this section, we will cover the two building blocks of the overall framework: 1) the encoder-predictor framework, and 2) the latent dynamic context model.

3.2.1 Encoder-Predictor Framework

Here, we describe a formulation of the encoder-decoder model discussed in Section 2.2 that shows how our representation of context can apply to a variety of models. We refer to the model as an encoder-predictor, because the predictor here mainly serves the purpose of providing training signal for unsupervised text representation learning.¹ Specifically, the model is described with two functions: the encoder maps inputs into intermediate variables, and the predictor maps the intermediate variables to outputs, where both the inputs and outputs are given during the training. Formally, we use the following definition

$$L(Y, d_{\theta}[e_{\gamma}(X, C_e), C_d]), \quad (3.1)$$

where X and Y are the input and output respectively, $e_{\gamma}(\cdot)$ and $d_{\theta}[\cdot]$ are the encoder and predictor functions, C_e and C_d represent the context information available to the encoder and predictor respectively, and $L(\cdot, \cdot)$ is the measurement for fit. As we are going to show, this formulation covers two models of interest, the RNN and the Seq2Seq model. In order to distinguish word-level models from sequence-level ones, we denote L^{word} as word-level model and L^{seq} as sequence-level model.

¹It would be interesting to explore whether the approach developed in this thesis can be used for controllable generation. However, that is beyond the scope of this thesis.

For RNNs, we assume there is an input sequence, $\mathbf{x}_1, \dots, \mathbf{x}_{t-1}$ and a vector \mathbf{h}_{t-1} encodes the sequential context, *i.e.* all inputs up to step $t - 1$. If the input unit is a word, then it is an RNN LM. The above formulation describes a standard RNN at the word-level with the overall objective as a summation of $L_t^{\text{word}}(\cdot, \cdot)$, $t = 1, \dots, T$, where at each time step t , $Y = \mathbf{x}_{t+1}$, $X = \mathbf{x}_t$, $C_e = \mathbf{h}_{t-1}$, $C_d = \phi$ and $e_\gamma(\cdot)$ and $d_\theta(\cdot)$ are shared by all time steps. Here, ϕ stands for no available context. If we additionally use the topic information encoded in vector \mathbf{z} which is inferred from an independent topic model for both the encoder and predictor context, *i.e.* $C_e = \{\mathbf{h}_{t-1}, \mathbf{z}\}$, $C_d = \mathbf{z}$, the resulting RNN LM is a context-aware RNN LM [10].

Since the context-aware RNN will be an important building block, we formally define the corresponding encoder and predictor, respectively, as

$$\mathbf{h}_t = f(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{i}_e), \quad (3.2)$$

$$\hat{\mathbf{x}}_{t+1} = g(\mathbf{h}_t, \mathbf{i}_d), \quad (3.3)$$

where \mathbf{i}_e and \mathbf{i}_d are the available encoder and predictor context vectors. In this case, both encoder and predictor context vectors are the topical information inferred by a topic model, *i.e.* $\mathbf{i}_e = \mathbf{i}_d = \mathbf{z}$. For a vanilla RNN LM which the context-aware RNN LM is built upon as in [10], $f(\mathbf{x}_t, \mathbf{h}_{t-1}, \phi) = \delta(\mathbf{W}[\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_1)$, and $g(\mathbf{h}_t, \phi) = \sigma(\mathbf{U}\mathbf{h}_t + \mathbf{b}_2)$ where $[\cdot, \cdot]$ is the concatenation operation, δ is the sigmoid function, σ is the softmax function, \mathbf{b}_1 and \mathbf{b}_2 are biases, \mathbf{W} and \mathbf{U} are the weight matrices.

Similarly, the overall objective of the vanilla Seq2Seq model in [22] can be described as a single L^{seq} , where $X = \mathbf{u}_1, \dots, \mathbf{u}_J$ the source language sequence of words, $Y = \mathbf{w}_1, \dots, \mathbf{w}_T$ the corresponding target language output sequence of words, $C_e = \phi$ and $C_d = \phi$. If there is additional user identity information as context for the predictor as in [35], it is a context-aware Seq2Seq model, *i.e.* a sequence-level analogy to the context-aware RNN LM. Different from the inferred topical context information in [10], the user identity context information used in [35] only includes static user demographic information.

It is easy to see the major difference between the aforementioned models: they operate at

different time scales, or input/output units. For a word-level model, the encoder consumes a new word and the predictor performs the prediction of the next word. For a sequence-level model, the encoder distills the input sequence information and the predictor predicts the next sequence. The RNN LM is the word-based model operating on a sequence of words covering multiple time steps. The Seq2Seq is the sequence-level model with only one input and output pair, *i.e.* a single step. Although we mainly take RNN-based models for example, any model that can operate on variable length inputs can be used as encoder and/or predictor. In the literature, popular models are RNNs, CNNs, Transformers [17], *etc.*

An important implication of this unifying view is that there is a hierarchical way of composing those models which corresponds to different factored context resulting in various types of contextualized representations. In this thesis, we are particularly interested in the following two compositions. First, two word-level models can be used to compose a sequence-level model. Specifically, a Seq2Seq model can be decomposed into one RNN LM for the input sequence and another RNN LM for the output sequence, such as a comment-response pair. Secondly, a sequence-level model can build upon another sequence-level model. Specifically, a sequence-level RNN can build upon another sequence-level RNN to characterize the sequence of user turns in a dialogue [89].

So far, we have presented a unifying view of the word-level RNN, the sequence-level RNN and the Seq2Seq model and their hierarchical compositions under the encoder-predictor framework. Next, we are going to talk about the modeling of the two context variables defined in Equation 3.1.

3.2.2 *Latent Dynamic Context Model*

For an RNN model, the context information can be represented using either one static variable or several dynamic variables. The former approach has been widely used in the literature for adapting the conditional models, such as domain adaptation and (controllable) text generation. However, sub-sequences (segments) within a sequence usually express dynamic multi-faceted information, such as intent, topic and style. Therefore, in this thesis, the global mode context is modeled with dynamically changing variables in an online fashion, which better reflects the characteristics of interactive language.

Another key aspect of the conditional language generation defined here is the conditional information which can be either observed or latent. Most previous work on the conditional RNN generation model relies on the observed conditional information. This usually involves human annotations for discrete variables in the form of labels or descriptors. For latent conditional information, one approach is to apply a separately trained latent variable model. For example, topic information is inferred from an independent topic model and used as the conditional information for the RNN LM [10]. In this thesis, we aim at jointly learning latent variables with the conditional RNN. To achieve this, we cast the problem as dynamically querying a memory to retrieve latent information that better explains the future sequence to be predicted. First, at step t , the RNN hidden state \mathbf{h}_t will be used to query the memory which contains a set of K memory vectors, $\mathbf{m}_1, \dots, \mathbf{m}_K \in \mathbb{R}^m$ resulting in K association scores, $a_1, \dots, a_K \in \mathbb{R}^+$ and $\sum_k a_k = 1$. In general, there are two popular ways of computing association scores: the multi-layer perceptron [27] and the linear dot-product [17]. The specific choice is detailed in subsequent chapters. Then, the conditional information is represented as a mixture of the memory vectors,

$$\tilde{\mathbf{m}} = \sum_{k=1}^K a_k \mathbf{m}_k. \tag{3.4}$$

By using the conditional information for the predictor RNN, we can achieve joint learning through the signal from the predictor as well as potentially interpretable latent global mode context vectors.

In the following chapters, we consider two ways of using the latent dynamic context model. For comments in online discussions, they generally consist of long sequences of sentences spanning multiple different salient community language factors, including topics, dialogue acts and idiosyncrasies. In order to capture those community characteristics, the latent dynamic context model is used with a word-level RNN model. For dialogues that are segmented into turns, we apply the latent dynamic model with sequence-level RNNs to model characteristics associated with users.

Chapter 4

MULTI-PARTY ONLINE DISCUSSIONS

Massive user-generated content on social media has drawn interest in predicting community reactions in the form of virality [49], popularity [47, 48, 60, 61], community endorsement [53, 54], persuasive impact [50–52], etc. Many of these studies have analyzed content-agnostic factors such as submission timing and author social status, as well as language factors that underlie the textual content, e.g., the topic and idiosyncrasies of the community. In particular, there is an increasing amount of work on online discussion forums such as Reddit that exploits the conversational and community-centric nature of the user-generated content [50–54, 61, 90], which contrasts with Twitter where the author’s social status seems to play a larger role in popularity. This chapter focuses on the problem of predicting community endorsement in Reddit, using the karma score¹ as a readily available measure of community endorsement.

Some of the prior work on Reddit investigates specific linguistic phenomena (e.g. politeness, topic relevance, community style matching) using feature engineering to understand their role in predicting community reactions [51, 53]. In contrast, this work explores methods for unsupervised text embedding learning using a model structured so as to provide some interpretability of the results when used in comment endorsement prediction. The model aims to characterize the interdependence of a comment, its global context and subsequent responses that are characteristic of multi-party online discussions. Specifically, we propose a factored neural model with separate mechanisms for representing global context, comment content and response generation. By factoring the model, we hope unsupervised learning will pick up different components of interactive language in the resulting embeddings, which will improve prediction of community reactions.

As discussed in Chapter 2, text embeddings have achieved great success in many language

¹karma = #up-votes - #down-votes.

processing applications, using both supervised and unsupervised methods. Unsupervised learning, in particular, has been successful at different levels, including words [2], sentences [4], and documents [5, 91]. Studies have also shown that the learned embedding captures both syntactic and semantic functions of words [1, 3, 6, 15]. At the same time, embeddings are often viewed as uninterpretable – it is difficult to align embedding dimensions to existing semantic or syntactic classes. This concern has triggered attempts in developing more interpretable embedding models [92], which is also a goal of our work. We leverage the fact that the structure of the distributional context impacts what is learned in an unsupervised way and include multiple objectives for separating different types of context.

Here, we are interested in linking two types of context with corresponding language factors learned in the embedding space that may impact comment reception. First, conformity to the topic and the language use of the community tends to make the content better accepted [48, 61, 93]. Those global *modes* typically influence the author’s generation of local *content*. Second, characteristics of a comment can influence the *responses* it triggers. Clearly, questions and statements will elicit different responses, and comments directed at a particular discussion participant may prompt that individual to respond. Of more interest here are aspects of comments that might elicit minimal response or responses with different sentiments, which are relevant for eventual endorsement.

The primary contribution of this work is the development of a factored neural model to jointly learn these aspects of multi-party discussions from a large collection of Reddit comments in an unsupervised fashion. Extending the recent neural attention model [29], the proposed model can interpret the learned latent global modes as community-related topic and style. A comment-response prediction model component captures aspects of the comment that are response triggers. The multi-factored comment embedding is evaluated on the task of predicting the comment endorsement for three online communities different in topic trends and writing style. The representation of textual information using our approach consistently outperforms multiple document embedding baselines, and analyses of the global modes and response trigger subvectors show that the model learns common communication strategies in discussion forums.

The work presented in this chapter is mainly based on [94]. The predictive task is described in

detail in [54]. My contribution includes:

- The design of the unsupervised embedding learning model for online discussion based on the encoder-predictor framework discussed in Chapter 3;
- A significant portion of the development of the karma prediction task for Reddit; and
- Both quantitative evaluation and qualitative analysis of the proposed unsupervised embeddings.

4.1 Model Description

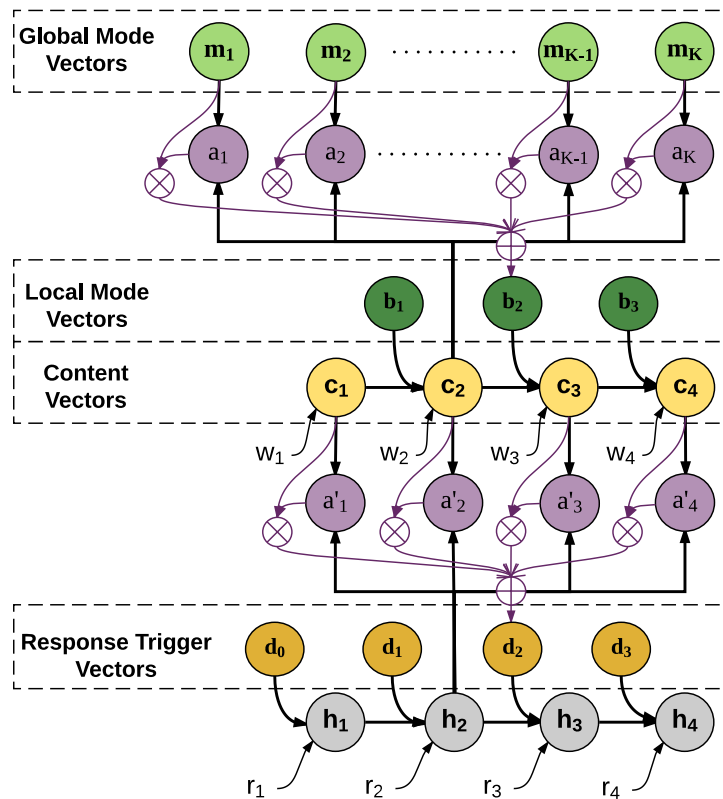


Figure 4.1: The structure of the full model omitting output layers, illustrating the computation of attention weights for b_2 and d_3 in a comment $w_{1:4}$ with its response $r_{1:4}$. Purple circles a_k and a'_j represent scalars computed in (4.2) and (4.8), respectively. \otimes and \oplus are scaling and element-wise addition operators, respectively. Black arrowed lines are connections carrying weight matrices.

To characterize different aspects of language use in a comment, the proposed model factorizes a comment embedding into two sub-vectors, *i.e.* a *local mode* vector and a *content* vector. The local mode vector, computed as a mixture of global mode vectors, exploits the global context of a comment. In the Reddit discussions that we use, the global mode represents the topic and language idiosyncracies (style) of a particular subreddit. More specific information communicated in the comment is captured in the content vector. The generation process of a comment is modeled through an RNN LM conditioned on local mode and content vectors, while the global mode vectors are jointly learned during the training.

In addition to the global context, the full model further exploits direct responses to the comment in order to learn better comment embeddings. This is achieved by modeling the prediction of comment responses through another RNN LM conditioned on *response trigger* vectors. The response trigger vectors are computed as mixtures of content vectors, with the idea that they will characterize aspects of the comment that incent others to respond, whether that be information or framing.

The full model is illustrated in Fig. 4.1. While the end goal is a joint framework, the model is described in the following two sub-sections in terms of two components: i) mode vectors for capturing global context, and ii) response trigger vectors for exploiting comment responses.

4.1.1 Mode Vectors

Using an RNN LM shown in the upper part of Fig. 4.1, we model the generation process of a word sequence by predicting the next word conditioned on the global context as well as the local content. The global context is encoded in the local mode vector, computed as a mixture of global mode vectors with mixture weights inferred based on content vectors. The local mode vector indicates where the comment fits in terms of what people in this subreddit generally say. It changes dynamically with the content vector as the comment generation progresses, considering the possibility of topic shifts and that the mode vectors may represent style as well as topic.

Suppose there is a set of K latent global modes with distributed representations $\mathbf{m}_{1:K} \in \mathbb{R}^n$.

For the t -th word w_t in a sequence, a local mode vector $\mathbf{b}_t \in \mathbb{R}^n$ is computed as

$$\mathbf{b}_t = \sum_{k=1}^K a(\mathbf{c}_t, \mathbf{m}_k) \otimes \mathbf{m}_k, \quad (4.1)$$

where $\mathbf{c}_t \in \mathbb{R}^n$ is the content vector for the current partial sequence $w_{1:t}$, \otimes multiplies a vector by a scalar, and the function $a(\mathbf{c}_t, \mathbf{m}_k)$ outputs a scalar *association probability* for the current content vector \mathbf{c}_t and a mode vector \mathbf{m}_k . The association function $a(\mathbf{c}, \mathbf{m}_k)$ is defined as

$$a(\mathbf{c}, \mathbf{m}_k) = \frac{\exp(\mathbf{v}^T \tanh(\mathbf{U} [\mathbf{c}; \mathbf{m}_k]))}{\sum_{i=1}^K \exp(\mathbf{v}^T \tanh(\mathbf{U} [\mathbf{c}; \mathbf{m}_i]))}, \quad (4.2)$$

where $\mathbf{U} \in \mathbb{R}^{n \times 2n}$ and $\mathbf{v} \in \mathbb{R}^n$ are parameters characterizing the similarity between \mathbf{m}_k and \mathbf{c} .

The computation of the association probability is the well-known attention mechanism [29]. However, unlike the original attention RNN model where the attended vector is concatenated with the input vector to augment the input to the recurrent layer, we adopt a residual learning approach [95] to learn content vectors. For the t -th word w_t in a sequence, the content vector \mathbf{c}_t under the original attention RNN model is computed as

$$\mathbf{c}_t = f(\mathbf{W}\mathbf{x}_t + \mathbf{G}\mathbf{b}_{t-1}, \mathbf{c}_{t-1}), \quad (4.3)$$

where $\mathbf{x}_t \in \mathbb{R}^d$ is the word embedding for w_t , $\mathbf{b}_{t-1} \in \mathbb{R}^n$ and $\mathbf{c}_{t-1} \in \mathbb{R}^n$ are previous local mode and content vectors, respectively, $\mathbf{W} \in \mathbb{R}^{n \times d}$ and $\mathbf{G} \in \mathbb{R}^{n \times n}$ are weight matrices transforming the input to the recurrent layer, and $f(\cdot, \cdot)$ is the recurrent layer activation function. To address the vanishing gradient issue in RNNs, we use the gated recurrent unit [27] for the RNN layer, *i.e.*

$$f(\mathbf{p}, \mathbf{q}) = (\mathbf{1} - \mathbf{u}) \odot \tanh(\mathbf{p} + \mathbf{R}[\mathbf{r} \odot \mathbf{q}]) + \mathbf{u} \odot \mathbf{q}, \quad (4.4)$$

where \odot is the element-wise multiplication, \mathbf{R} is the recurrent weight matrix, and \mathbf{u} and \mathbf{r} are the

update and reset gates, respectively. In this work, we compute the content vector \mathbf{c}_t as follows:

$$\mathbf{c}_t = f(\mathbf{W}\mathbf{x}_t, \mathbf{G}\mathbf{b}_{t-1} + \mathbf{c}_{t-1}). \quad (4.5)$$

Comparing (4.3) and (4.5), it can be seen that we first aggregate the local mode vector \mathbf{b}_{t-1} and the content vector \mathbf{c}_{t-1} and treat the resulting vector $\mathbf{G}\mathbf{b}_{t-1} + \mathbf{c}_{t-1}$ as the memory of the recurrent layer. The resulting hidden state vectors from the recurrent layer are content vectors \mathbf{c}_t 's. The use of residual learning is motivated by the following considerations. The local mode vector \mathbf{b}_{t-1} can be seen as a non-linear transformation of \mathbf{c}_{t-1} into a global mode space parameterized by $\mathbf{m}_{1:K}$. If the global information carried in \mathbf{b}_{t-1} is residual for generating the following word in the comment, the model only needs to exploit the information in local content \mathbf{c}_{t-1} and learns to zero out the local mode vector \mathbf{b}_{t-1} , *i.e.* $\mathbf{G} = 0$. Shown by [95], the residual learning usually leads to a more well-conditioned model which promises better generalization ability.

Finally, the RNN LM estimates the probability of the $(t + 1)$ -th word w_{t+1} based on the current local mode vector \mathbf{b}_t and content vector \mathbf{c}_t , *i.e.*

$$\Pr(w_{t+1}|w_{1:t}) = \text{softmax}(\mathbf{Q}(\mathbf{G}\mathbf{b}_t + \mathbf{c}_t)), \quad (4.6)$$

where $\mathbf{Q} \in \mathbb{R}^{V \times n}$ is the weight matrix, and V is the vocabulary size. Note that the model jointly learns all parameters in the RNN together with the mode vectors $\mathbf{m}_{1:K}$. This differentiates our model from the context-dependent RNN LM [96], which is conditioned on a context vector inferred from a pre-trained topic model.

4.1.2 Response Trigger Vectors

Another important aspect of comments in online discussions is how other participants react to the content. In order to exploit those characteristics, we use comment-reply pairs in online discussions and build this component upon the encoder-decoder framework with an attention mechanism [29], as illustrated in the lower part of Fig. 4.1. The decoder is essentially another RNN LM conditioned

on response trigger vectors aiming at distilling relevant parts of the comment which other people are responding to.

Let r_j denote the j -th word in a reply to a comment w_1, \dots, w_T . The decoder RNN LM computes a hidden vector $\mathbf{h}_j \in \mathbb{R}^n$ for r_j as follows,

$$\mathbf{h}_j = f(\mathbf{W}^\dagger \mathbf{x}_j + \mathbf{G}^\dagger \mathbf{d}_{j-1}, \mathbf{h}_{j-1}), \quad (4.7)$$

where $\mathbf{W}^\dagger \in \mathbb{R}^{n \times d}$ and $\mathbf{G}^\dagger \in \mathbb{R}^{n \times n}$ are weight matrices, \mathbf{x}_j is r_j 's word embeddings from a shared embedding dictionary as used by the encoder RNN LM in Subsection 4.1.1, and $\mathbf{d}_{j-1} \in \mathbb{R}^n$ and $\mathbf{h}_{j-1} \in \mathbb{R}^n$ are the response trigger vector and hidden vector at the previous time step, respectively. The initial hidden vector \mathbf{h}_0 is set to be the last content vector \mathbf{c}_T . With the comment's content vectors $\mathbf{c}_1, \dots, \mathbf{c}_T$ obtained from the encoder RNN LM in Subsection 4.1.1, a response trigger vector \mathbf{d}_j is computed as the mixture:

$$\mathbf{d}_j = \sum_{t=1}^T a'(\mathbf{h}_j, \mathbf{c}_t) \cdot \mathbf{c}_t, \quad (4.8)$$

where $a'(\mathbf{h}_j, \mathbf{c}_t)$ is a similar function to $a(\mathbf{c}_t, \mathbf{m}_k)$ defined in (4.2) with different parameters. Similar to the encoder RNN LM, the decoder RNN LM estimates the probability of the $(j+1)$ -th word r_{j+1} in the reply based on the hidden vector \mathbf{h}_j and the response trigger vector \mathbf{d}_j , *i.e.*

$$\Pr(r_{j+1}|r_{1:j}) = \text{softmax}(\mathbf{Q}^\dagger [\mathbf{h}_j; \mathbf{d}_j]),$$

where $\mathbf{Q}^\dagger \in \mathbb{R}^{V \times 2n}$ is the weight matrix.

4.2 Model Learning

The full model is trained by maximizing the log-likelihood of the data, *i.e.*

$$\sum_i \left(\log \Pr(w_{1:T(i)}^{(i)}) + \alpha \log \Pr(r_{1:J(i)}^{(i)} | w_{1:T(i)}^{(i)}) \right),$$

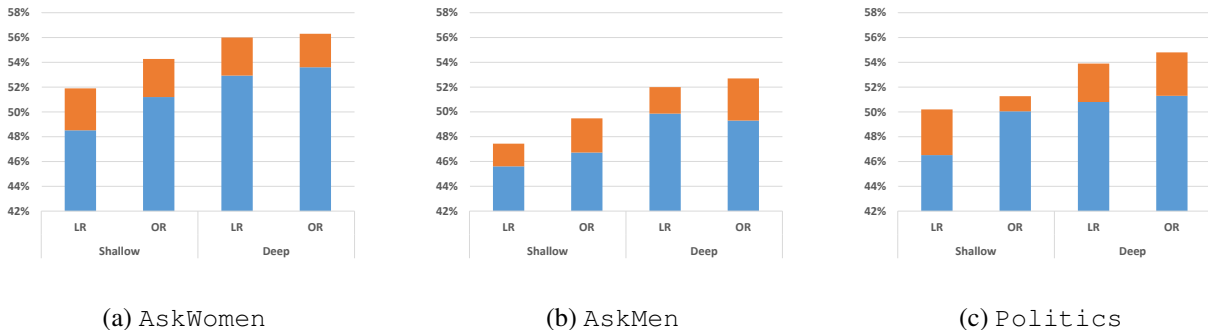


Figure 4.2: Averaged F1 scores of different classifiers. Blue bars show the performance using no comment embeddings. Orange bars show the absolute improvement by using factored comment embeddings.

where the two terms correspond to the log-likelihood of the encoder RNN LM and the decoder RNN LM, respectively, and α is the hyper parameter which weights the importance of the second term. In our experiments, we let $\alpha = 0.1$. During the training, each comment-reply pair $(w^{(i)}, r^{(i)})$ is used as a training sample. Considering that comments may receive a huge number of replies, we keep up to 5 replies for each comment. We use the first 50 words of comments and the first 20 words of replies. If a comment has no reply, a special token is used. All weights are randomly initialized according to $\mathcal{N}(0, 0.01)$. The model is optimized using Adam [97] with an initial learning rate 0.001. Once the validation log-likelihood decreases for the first time, we halve the learning rate at each epoch. The training process is terminated when the validation log-likelihood decreases for the second time. In our experiments, we learn word embeddings of dimension $d = 256$ from scratch. The number of modes K is set to 16. A single-layer RNN is used, with the dimension n of hidden layers set to 64.

4.3 Data and Task Details

Data: In this chapter, we work with Reddit discussion threads, taking advantage of their conversational and community-centric nature as well as the available karma scores. Each thread starts from a post and grows with comments to the post or other comments within the thread, presented as a

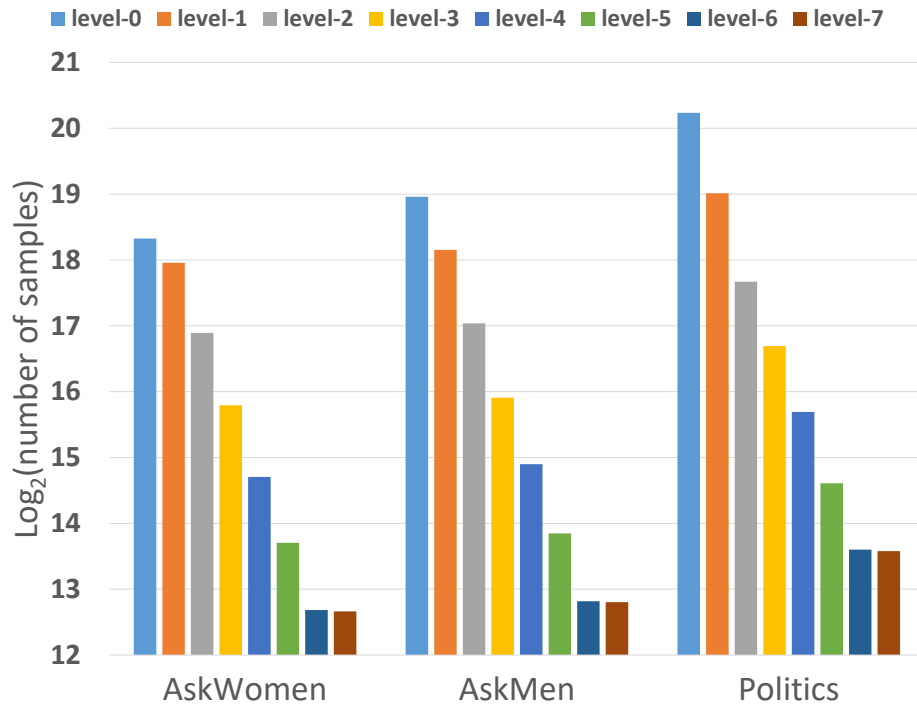


Figure 4.3: The quantized karma distribution.

tree structure. Posts and comments can be voted up or down by readers depending on whether they agree or disagree with the opinion, find it amusing vs. offensive, etc. A *karma* score is computed as the difference between up-votes and down-votes, which has been used as a proxy of community endorsement for a Reddit comment. In this work, we use three popular subreddits with different topics and styles are studied: *AskWomen* (814K comments), *AskMen* (1,057K comments), and *Politics* (2,180K comments). The label distributions are shown in Fig. 4.3. For each subreddit, we randomly split comments by threads into training, validation, and test data, with a 3:1:1 ratio.

Task: Considering the heavy-tailed Zipfian distribution of karma scores, regression with a mean squared error objective may not be informative because low-karma comments dominate the overall objective. Following [54], we quantize comment karma scores into 8 discrete levels and design a task consisting of 7 *binary* classification subtasks which individually predict whether a comment’s

karma is *at least* level- l for each level $l = 1, \dots, 7$. This task is sensitive to the order of quantized karma scores, e.g., for the level-6 subtask, predicting a comment as level-5 or level-7 would lead to different evaluation results such as recall, which is not the case for a standard multi-class classification task. Additionally, compared to a standard multi-class classification task, these subtasks alleviate the unbalanced data issue.

Evaluation metric: For each level- l binary classification subtask, we compute the F1 score by treating comments at levels lower than l as negative samples and others as positive samples. Note that we only compute F1 scores for $l \in \{1, \dots, 7\}$ since no comment is at a level lower than 0. The averaged F1 scores is used as an indicator of the overall prediction performance.

4.4 Experiments

We evaluate the effectiveness of the factored comment embeddings on the quantized karma prediction task. We use the concatenation of the local mode vector and the content vector at the last time step as the factored comment embedding. Note that for the task we only have access to the current comment, so we cannot include the response trigger vectors. However, the use of response generation in the training objective lead to content vectors that are predictive of response characteristics. First, we study the overall prediction performance of four different classifiers under two settings, i.e., using factored comment embeddings or not. Then we compare the factored comment embeddings inferred from the full model and its two variants with other kinds of text features using the best type of classifiers. Finally, we carry out error analysis on prediction results of the best classifiers using the factored comment embeddings.

4.4.1 Classifiers

The following four types of classifiers are studied:

- **ShallowLR:** A standard multi-class logistic regression model;
- **ShallowOR:** An ordinal regression model [98], which can exploit the order of the quantized karma labels;
- **DeepLR:** A feed-forward neural network using the logistic regression objective;

| | AskWomen | AskMen | Politics |
|-------------------|--------------|--------------|--------------|
| Baseline | 53.6% | 49.3% | 51.3% |
| BoW | 53.1% | 50.9% | 51.8% |
| LDA | 55.3% | 51.1% | 52.5% |
| Doc2Vec | 55.2% | 51.7% | 53.0% |
| Factored\M | 54.2% | 51.8% | 52.9% |
| Factored\R | 55.1% | 51.9% | 53.4% |
| Factored | 56.3% | 52.7% | 54.8% |

Table 4.1: Averaged F1 scores of DeepOR classifiers using different text features. Baseline results do not use any text features.

- **DeepOR**: A feed-forward neural network using the ordinal regression objective.

These classifiers have different objectives and model complexities, allowing us to study the robustness of the learned comment embeddings. As a text-free baseline, we train the classifiers using only content-agnostic features, e.g., timing and discussion tree structure, which have strong correlations with community endorsement [53, 54]. All classifiers are trained on the training data for each subreddit independently, with hyper-parameter tuned on the validation data.

The prediction performance on the test data is shown in Fig. 4.2. We observe that using comment embeddings consistently improves the performance of these classifiers. While ShallowOR significantly outperforms ShallowLR, indicating the usefulness of exploiting the order information in quantized karma labels, the difference is much smaller for deep classifiers. Also, deep classifiers consistently outperform their shallow counterparts.

4.4.2 Text Features

We compare the factored comment embeddings with the following text features:

- **BoW**: A sparse bag-of-word representation;
- **LDA**: A vector of topic probabilities inferred from the topic modeling [99];
- **Doc2Vec**: Embeddings inferred from the paragraph vector model [5].

In addition to the factored comment embeddings obtained from our full model, we study two variants of the full model: 1) a model trained without the mode vector component (**Factored\M**),

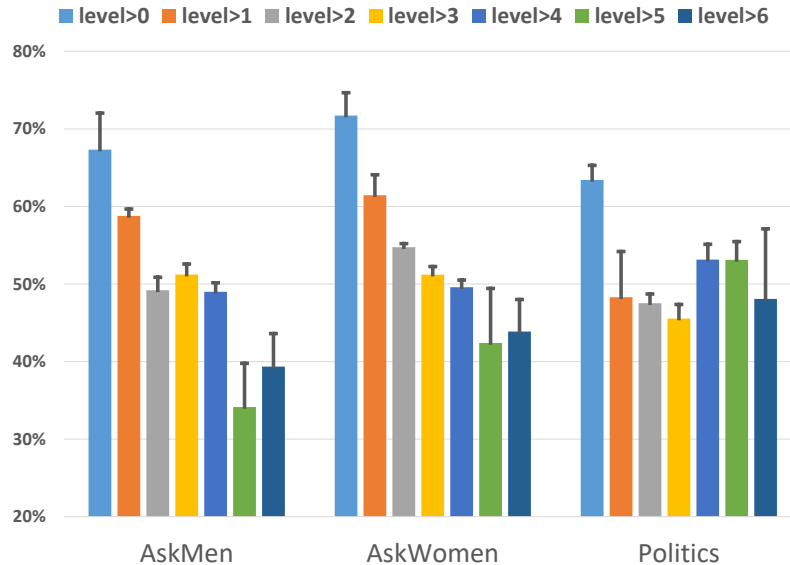


Figure 4.4: F1 scores of the DeepOR classifier for individual subtasks. Error bars indicate the improvement of using the factored comment embeddings over the classifier using no text features.

which is a normal sequence-to-sequence attention model [29], and 2) a model trained without the response trigger vector component (**Factored**\R). Since the DeepLR and the DeepOR perform best across all subreddits and they have similar trends, we report results of the DeepOR in Tabel 4.1.

Among all text features, the BoW has the worst averaged F1 scores and even hurts the performance for AskWomen, probably due to the data sparsity problem. Both the LDA and the Doc2Vec outperform the BoW. The Doc2Vec performs slightly better on AskMen and Politics, which might be attributed to the relative larger training data size. The factored comment embeddings derived from the full model consistently achieve better averaged F1 scores. It can be observed that the two variants of the full model mostly lead to similar performance as Doc2Vec, though the Factored\R embeddings usually have higher averaged F1 scores than the Factored\M embeddings. These results suggest advantages of jointly modeling two components, which may drive the model to discover more latent factors and patterns in the data that could be useful for downstream tasks.

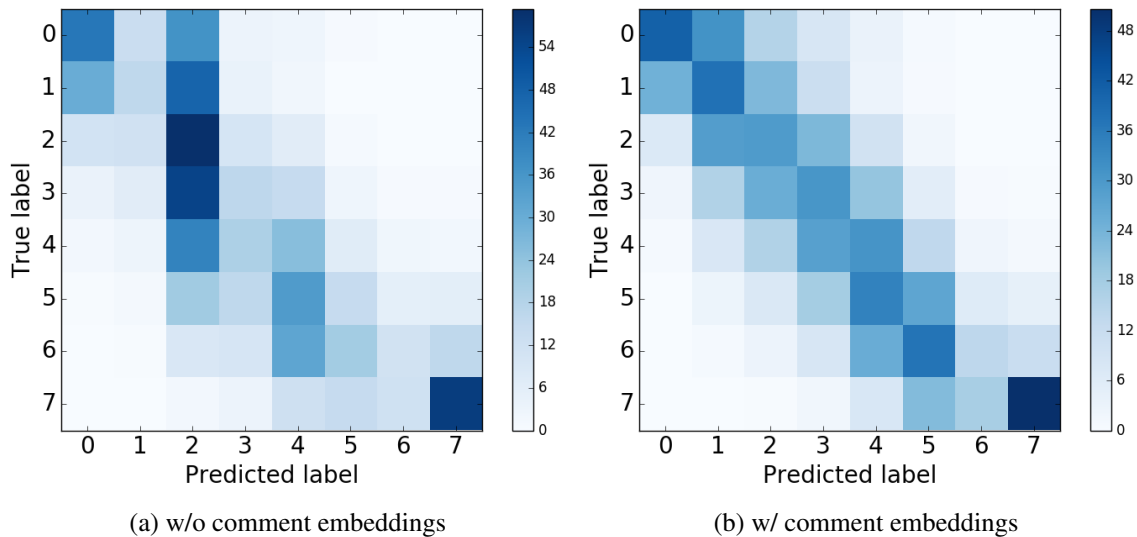


Figure 4.5: The confusion matrices for the DeepOR classifier on `Politics`. The color of cell on the i -th row and the j -th column indicates the percentage of comments with quantized karma level i that are classified as j , and each row is normalized.

4.4.3 Error Analysis

In this subsection, we focus on analyzing how factored comment embeddings improve the prediction results of the DeepOR classifiers. The F1 scores for individual subtasks are shown in Fig. 4.4. Note that the higher the level is, the more skewed the task is, *i.e.* a lower positive ratio. As expected, comments with the lowest endorsement level are easier to classify. Adding comment embeddings primarily boosts the performance of the classifier on the high-endorsement tasks (level > 5) and the low-endorsement tasks (levels 0, 1).

Confusion matrices for the DeepOR classifier with and without factored comment embeddings are shown in Fig. 4.5 for `Politics`. Using the additional comment embeddings leads to a higher concentration of cell weights near the diagonals, corresponding to errors that mainly confuse neighboring levels. Without any text features, the classifier seems to only distinguish four levels. We observe similar trends on `AskWomen` and `AskMen`.

4.5 Qualitative Analysis

In this section, we conduct analysis to better understand what the factored model is learning, again using the `Politics` subreddit. First, we analyze latent global modes learned from the full model. For each global mode, we extract comments with top association scores. Note that the model assumes a locally coherent mixture of global modes and updates the mixture for each observed word. Thus, each comment receives a sequence of association probabilities over the global modes. The association score β_k between a comment $w_{1:T}$ and `Mode- k` is then computed as $\beta_k = \max_{t \in \{1, \dots, T\}} a(\mathbf{c}_t, \mathbf{m}_k)$ for $k \in \{1, \dots, K\}$, where $a(\mathbf{c}_t, \mathbf{m}_k)$ is defined in (4.2). In Table 4.2, we show examples from the most coherent modes out of the 16 learned modes. Some modes seem to be capturing style (modes 2, 6, and 10), while others are related to topics (modes 7 and 16). `Mode-2` captures the style of starting with a rhetorical question to express negative sentiment and disagreement. Many comments in `Mode-6` begin with words of drawing attention such as “bull” and “psst”. `Mode-10` tends to be associated with comments telling a story about a closely related person. Many comments in `Mode-7` discuss low salaries, whereas `Mode-16` comments discuss politicians or ideology of the Republican.

The characteristics of examples in modes 2 and 6 suggested that modes might have a location dependency, so we looked at word positions with the strongest association of each mode, *i.e.* $\operatorname{argmax}_{t \in \{1, \dots, T\}} a(\mathbf{c}_t, \mathbf{m}_k)$. For each `Mode- k` , we only keep comments with association score higher than $\operatorname{mean}(\beta_k) + \operatorname{std}(\beta_k)$. Fig. 4.6 shows the box plot of locations where the strongest association happens. It can be seen that modes 2 and 6 usually have the strongest association at the beginning of a comment. For modes 3, 8, 15 and 16, the strongest associations occur over a wider span in comments. In addition to the interpretability of the learned modes, as one can get from LDA, these observations suggest that our model may further capture word location effects which may help with predicting community endorsement.

Next, we analyze the response characteristics by examining the response trigger vectors associated with the onset of comment responses, which is a special start-of-reply token. These response trigger vectors are clustered into 8 classes via k-means and visualized in Fig. 4.7 using t-SNE [100].

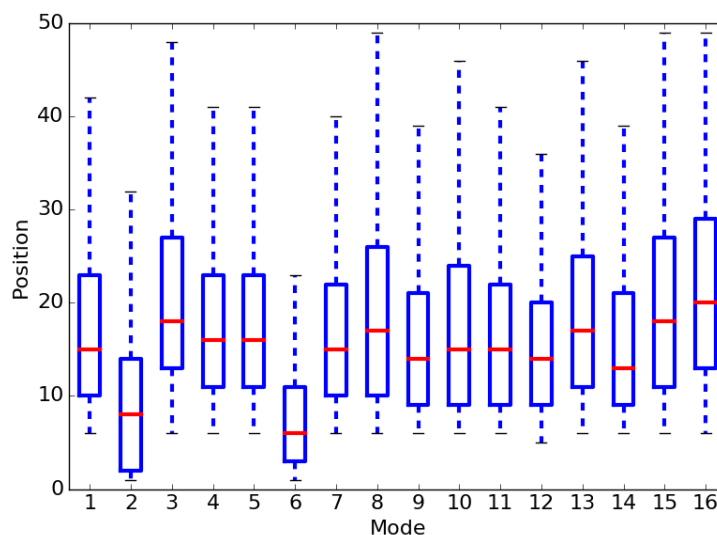


Figure 4.6: The box plot of strongest association positions for each global mode in `Politics`.

For each cluster, we study the karma distribution, as well as comments together with the first reply.

The horizontal dimension seems to be associated with how many replies a comment elicits. The vertical dimension is less interpretable but most clusters have identifiable traits. The far left classes (Class-1&4) both have few replies and low karma, often two-party exchanges where Class-4 has more negative sentiment. Class-2 comments tend to involve complements, whereas comments in Class-3 usually trigger a reply with a *but*-clause for a contrast and disagreement intent. Comments in Class-5 mostly receive responses expanding on the original comments. Class-6 has a lot of sarcastic and cynical comments and replies. Comments in Class-7 are mostly anomalous since their first responses were usually “[deleted]”, *i.e.* the responses are removed either by the authors themselves or administrators. It seems there are multiple response trigger factors in the proposed embedding model, some may reflect dialog acts and others sentiment, any of which may be helpful in predicting community endorsement.

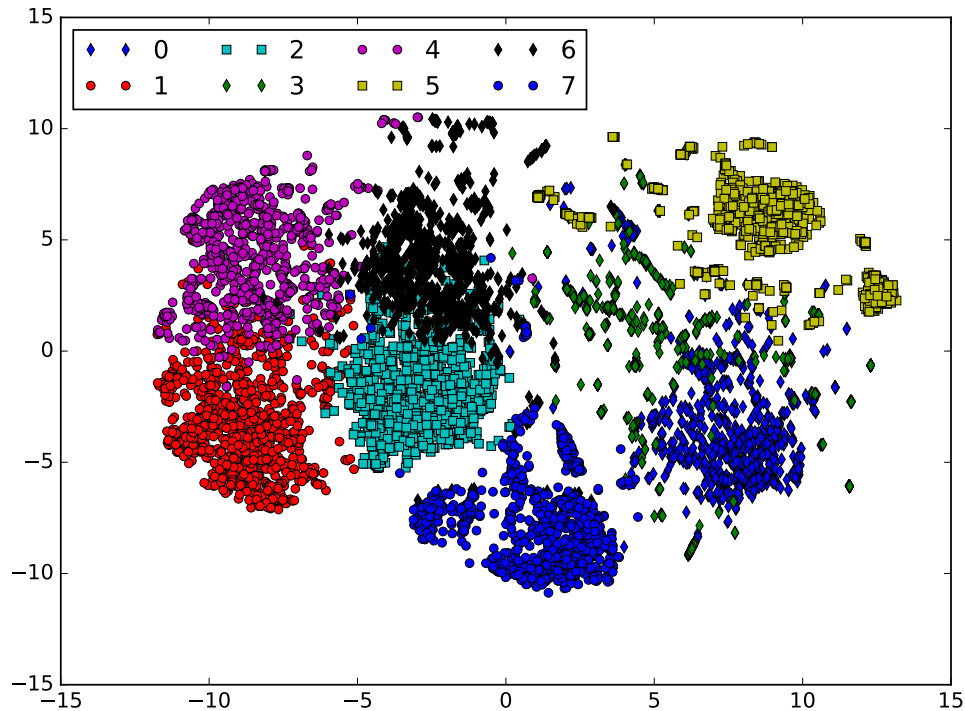


Figure 4.7: t-SNE visualization of response trigger vectors clustered using k-means.

4.6 Summary

In this chapter, we introduce a new factored neural model for unsupervised learning of comment embeddings leveraging two different types of context in online discussions. By extending the attention mechanism, our method is able to jointly model global context, comment content and response generation.

Quantitative experiments on three different subreddits show that the factored embeddings achieve consistent improvement in predicting quantized karma scores over other standard document embedding methods. Analyses of the learned global modes show community-related style and topic characteristics are captured in our model. Also, we observe that response trigger vectors characterize certain aspects of comments that elicit different response patterns.

| | |
|---------|--|
| Mode-2 | <ul style="list-style-type: none"> • Oh come on! Really? One can't make that trip and spend maybe half and save the other for milk, bread and things that do spoil? ... • Remind me. How many filibusters did Harry Reid conduct this year? ... • Feckless tyrant? How did you do that with your brain? ... • Seriously? You have to be registered to vote. ... • Holy f*, seriously? This is some heavy duty shit. ... |
| Mode-6 | <ul style="list-style-type: none"> • Bull. Plenty of individuals influence policy by never missing a single chance to vote, no matter how minor the election. ... • Bull. Conservatives hate Obamacare so much because if their constituents got mental health treatment, they'd stop voting Republican. • Utter bull s*. Where was the compromise from Obama and the Dems when they pushed through Obamacare without ONE Republican vote. ... • psst... it's college • psst- he's "black" - meaning that one of his ancestors is black (as if it's pollutant of some sort). |
| Mode-7 | <ul style="list-style-type: none"> • ... , I used to work 55+ hours a week, salaried, lower quartile salary to boost. ... • Or possibly that the standard of living between unemployment and the "jobs" that are out there is really insignificant. ... • Where on earth is 7.25 a living wage? If by some miracle you get 40 hours a week that's only \$1,160 before taxes. ... • If you have to work 40 hours a week to pay your bills that means you are controlled in your fight for survival. ... • ... Working 15 hours a week for extra pocket money when you're a teen is easy. Working 50 hours a week at fast food to cover rent, food, ... |
| Mode-10 | <ul style="list-style-type: none"> • ... Had a guy stalk a trans friend of mine for months trying to terrorize her because ... • A co-worker of mine got audited by the IRS because ... • ... Some conservative friends of mind wanted to meet up at a coffee house with shittier coffee because the other one was too "liberal". ... • ... Friend of mine works with mentally unstable and aggressive people as part of some social service. ... • ... A student of mine asked our own AP about an atheist group and he just flat out said "You kidding me?" ... |
| Mode-16 | <ul style="list-style-type: none"> • ... These same people will continue to listen to the bullshit that is the Republican Party. And when that happens, they have this twisted reality ... • ... After spending almost my entire life in Texas and as a Born gain evangelical conservative Republican, I learned my lessons about how completely dishonest and corrupted that entire culture is the hard way. ... I will never gain ever vote for or support any kind of conservative. ... • ... has been our greatest embarrassment, but what makes matter even worse is the support he has for re-election. I would not be surprised ... • Well, it is entirely possible that ... the underlying cause of Limbaugh's attack was that this guy was playing the type of dirty politics ... • ... this was more of a referendum on the GOP leadership in Congress by Republican voters, because let's face it, they haven't done anything... |

Table 4.2: Examples of comments associated with the learned global modes for *Politics*.

Chapter 5

OPEN-DOMAIN SOCIAL-CHAT DIALOGUES

Representing language in context is key to improving 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. In this chapter, we focus on the problem of modeling the speaker. Accounting for author/speaker variations has been shown to be useful in many NLP tasks, including language understanding [46, 79], language generation [35, 80], human-computer dialog policy [101], query completion [102, 103], comment recommendation [104] 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 [4, 7, 8, 105, 106], our approach is to use unsupervised learning with a neural model of a speaker’s dialog history. The model uses latent global mode vectors for representing a speaker turn as in Chapter 4, which again 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, based on the encoder-predictor and the

latent dynamic context model discussed in Chapter 3, we develop a 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 embeddings 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 state-of-the-art results.

The work presented here is published in [107]. My contribution includes:

- The design and implementation of the unsupervised speaker model for two-party open-domain social-chat dialogues based on the encoder-predictor framework discussed in Chapter 3;
- A significant portion of the development of the social-bot dialogue system described in [108, 109], and introduction of the topic decision prediction task for social-bot dialogues; and
- Both quantitative evaluation and qualitative analysis of the proposed speaker model.

5.1 *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

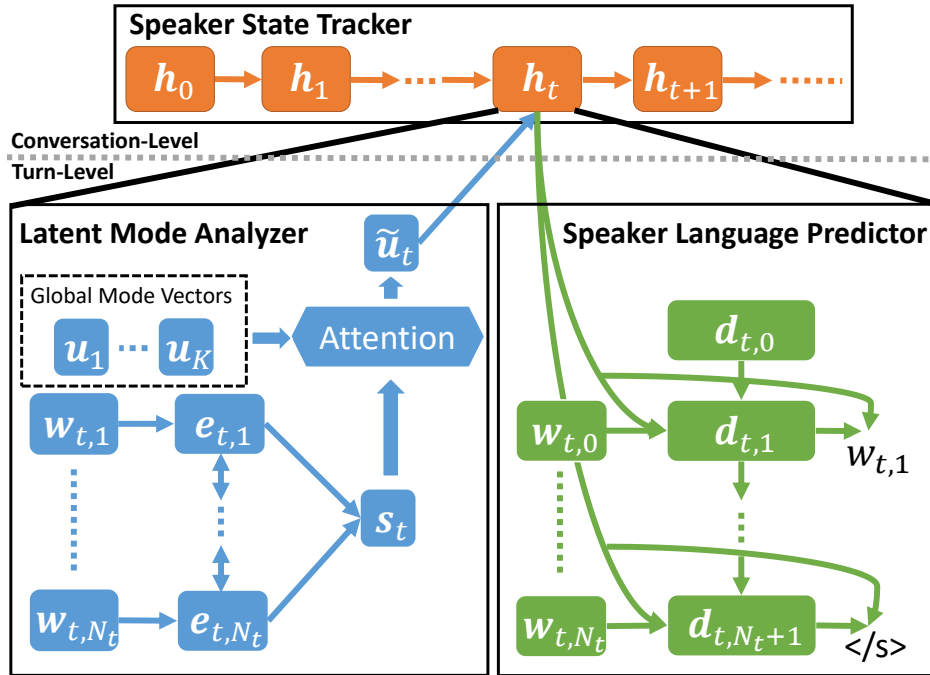


Figure 5.1: The dynamic speaker model. The speaker state tracker operates at the conversation level. The latent model analyzer and speaker language predictor operate at the turn level. The figure only shows processes in those two components for the turn t .

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. 5.1. 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 mode 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 [26]. In our experiments, the three components are trained jointly.

5.1.1 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 $w_{t,1}, \dots, w_{t,N_t}$ 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 $\mathbf{e}_{t,n}^F = g^F(\mathbf{w}_{t,n}, \mathbf{e}_{t,n-1}^F) \in \mathbb{R}^m$ for $n = 1, \dots, N_t$, where $g^F(\cdot, \cdot)$ denotes the forward LSTM function. The backward LSTM computes its hidden states $\mathbf{e}_{t,n}^B \in \mathbb{R}^m$ similarly. The initial hidden states $\mathbf{e}_{t,0}^F$ and \mathbf{e}_{t,N_t+1}^B are set to zeros. The summary vector \mathbf{s}_t is the concatenation of the two final hidden states, $\mathbf{s}_t = [\mathbf{e}_{t,N_t}^F, \mathbf{e}_{t,1}^B]$.

In the second stage, the utterance summary vector \mathbf{s}_t is compared with K global mode vectors $\mathbf{u}_1, \dots, \mathbf{u}_K \in \mathbb{R}^c$ 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 [17] as follows,

$$a_{t,k} = \frac{\exp(\langle \mathbf{P}\mathbf{s}_t, \mathbf{Q}\mathbf{u}_k \rangle)}{\sum_{k'=1}^K \exp(\langle \mathbf{P}\mathbf{s}_t, \mathbf{Q}\mathbf{u}_{k'} \rangle)}, \quad (5.1)$$

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 $\tilde{\mathbf{u}}_t = \sum_{k=1}^K a_{t,k} \mathbf{u}_k$.

5.1.2 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 $\tilde{\mathbf{u}}_{t,1}, \dots, \tilde{\mathbf{u}}_{t,N_t}$. 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 .

5.1.3 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 $n = 1, \dots, N_t + 1$, 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

$$\Pr(w_{t,n}|w_{t,<n}) = \text{softmax}(\mathbf{V}r^O(\mathbf{h}_t, \mathbf{d}_{t,n})), \quad (5.2)$$

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.

5.1.4 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 (5.2). The Adam optimizer [97] 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 [110] 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.

5.2 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 [59] from the Sounding Board system [109, 111]. 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.

5.2.1 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. A sample conversation is shown in Table 5.1, which is a combination of conversation segments from real users. In the given example, there are five topics: `Superman` (turn 5), `science` (turn 13), `Henry Cavill` (turn 8), `movie Superman` (turn 10) and `car` (turn 14), where the first two are user-requested and the last three are bot-suggested. In this case, the user accepts two topics, `Henry Cavill` and `car`, but rejects one topic, `movie Superman`.

In socialbot conversations, a system turn is always followed by a user turn and vice versa. We tag system turns that involve 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 in this subset of conversations. Not every system turn involves 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 5.2. 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.

5.2.2 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 | Speaker | Utterance |
|------|---------|--|
| 1 | User | <i>let's chat</i> |
| 2 | Bot | <i>Hi! This is an Alexa Prize socialbot. How's your day?</i> |
| 3 | User | <i>I'm [five]</i> |
| 4 | Bot | <i>Glad to hear it! We could talk about robots, science, or sports.</i> |
| 5 | User | <i>Superman</i> |
| 6 | Bot | <i>It looks like you wanna chat about Superman. I was high up in the cloud when I realized: If people don't recognize Clark Kent as Superman because of his glasses, does his eye doctor know his true identity?</i> |
| 7 | User | <i>I guess so</i> |
| 8 | Bot | <i>Did you know that Henry Cavill almost missed the call for the role of Superman cause he was playing "World of Warcraft"?</i> |
| 9 | User | <i>really I didn't know that</i> |
| 10 | Bot | <i>Weird, right? Speaking of this, did you watch the movie Superman? It was released in 1997. It's a comedy.</i> |
| 11 | User | <i>can we about something else i want to talk about science</i> |
| 12 | Bot | <i>I love it! Did you see this article from Last Friday? The title was: Deep learning neural networks power self-driving cars, predictive policing, and medical diagnoses.</i> |
| 13 | User | <i>no i didn't</i> |
| 14 | Bot | <i>Okay. This news mentioned car. Would you like to hear some advice about that?</i> |
| 15 | User | <i>yes</i> |
| 16 | Bot | <i>Great! My friend gave me this advice. If you get into an accident, try to take note of the plate of at least one other car which might have seen it occur.</i> |
| 17 | User | <i>that's true thanks stop</i> |

Table 5.1: A sample conversation between the user and Sounding Board. There are two user-requested topics: Superman (turn 5), science (turn 11). There are three bot-suggested topics: Henry Cavill (turn 8), movie Superman (turn 10) and car (turn 14). A suspected speech recognition error in the user utterance at turn 3 is shown in [].

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

Table 5.2: Data statistics of the topic decision dataset.

turn t , which precedes t' . The task is to predict whether user will accept the topic in the next turn.

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 pre-trained 300-dimensional word embeddings [110].

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 [112], we derive a static user embedding vector for a conversation using the first 5 user turns and apply it for predicting all 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 [97]. The training protocol is similar to that described in §5.1.4. 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 oper-

ating 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.

5.2.3 Experiments: Static User Embeddings

As described in §5.2.2, we use the first 5 user turns to derive the user embedding vector for a conversation. We compare our dynamic speaker model used in static mode 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 $\tilde{\mathbf{u}}_t$ 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 for all subsequent turns.

UtteranceLDA: The latent Dirichlet allocation (LDA) model [99] is trained with 16 latent groups by treating all user utterances in a conversation as a document.¹ 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 [113]. 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 §5.1.1. The auto-encoder is trained on all user utterances in the training data, using the same training protocol described in §5.1.4. We set the hidden layer size to 128. The user embedding vector is constructed by applying the max-pooling operation over the summary vectors $\mathbf{s}_1, \dots, \mathbf{s}_5$ 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-

¹To 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.

| 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 5.3: Test set results (in %) for topic decision predictions using *static* user embeddings.

| 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 5.4: 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 ($p < .001$).

dimensional embeddings [110] for words in the topic.

Experiment results are summarized in Table 5.3. 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.

5.2.4 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 §5.2.3.

UtteranceAE+LSTM: This model uses an LSTM to encode the summary vector sequence derived from the same utterance auto-encoder used in §5.2.3. 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 §5.2.3. Since not every user turn is associated with a topic 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 TopicDecision-LSTM, 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 5.4. The DynamicSpeakerModel performs the best. Comparing to results in Table 5.3, 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 significant ($p < .001$). Predictions from UtteranceAE + LSTM and DynamicSpeakerModel are also significantly different based on both tests ($p < .001$).

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

5.2.5 Qualitative Analysis

First, we manually inspect the predictions from the TopicDecisionLSTM and DynamicSpeakerModel used in §5.2.4 and the static baseline TopicPrior in §5.2.3. 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 5.5, 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 (5.1). For each mode, we list top associated user utterances in Tables 5.6–5.7. For modes learned in

| | |
|--------|--|
| Mode-0 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • gotcha • hiya • possibly |
| Mode-2 | <ul style="list-style-type: none"> • serious • are you serious • that is a paradox |
| Mode-3 | <ul style="list-style-type: none"> • alexa resume pandora • alexa connect bluetooth • no bye bye alexa |
| Mode-4 | <ul style="list-style-type: none"> • that is fascinating • whoa • that that's cool |
| Mode-5 | <ul style="list-style-type: none"> • i did not that's not surprising • i did not i did not knew that • unfortunately |
| Mode-6 | <ul style="list-style-type: none"> • somewhat • yes yes yes yes yes • yes i did it was on the news |
| Mode-7 | <ul style="list-style-type: none"> • mhm • ok • fascinating |

Table 5.6: User utterances in socialbot conversations that have top association scores for individual latent modes: modes 0–7.

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.

| | |
|---------|---|
| Mode-8 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • meow • award shows • celebrity |
| Mode-10 | <ul style="list-style-type: none"> • no let's talk about butterflies • no let's talk about snakes • can we talk about kardashians |
| Mode-11 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • ten million • thirty percent • what's p. r. |
| Mode-14 | <ul style="list-style-type: none"> • dog • dogs • tv |
| Mode-15 | <ul style="list-style-type: none"> • now • not now |

Table 5.7: User utterances in socialbot conversations that have top association scores for individual latent modes: modes 8–15.

5.3 *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 [114, 115]. In this section, we apply the proposed dynamic speaker model to the dialog act classification task.

5.3.1 *Data*

We use the Switchboard Dialog Act Corpus (SwDA), which has dialog act annotations on two-party human-human speech conversations [116, 117]. In total, there are 1155 open-domain conversations with manual transcripts. Following recent work [118–120], we use 1115 conversations for training, 19 for testing, and the rest 21 for development.² The original fine-grained dialog act labels are mapped to 42 classes.³ 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 [8, 106], 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 [121], which (like Switchboard) consists of two-party human-to-human telephone conversations but without annotations for dialog acts.

5.3.2 *Dialog Act Tagging Model*

We use an attention-based LSTM tagging model for the dialog act classification. As shown in Fig. 5.2, 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

²The training and test split files are downloaded from <https://web.stanford.edu/~jurafsky/ws97/>.

³Dialog 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.

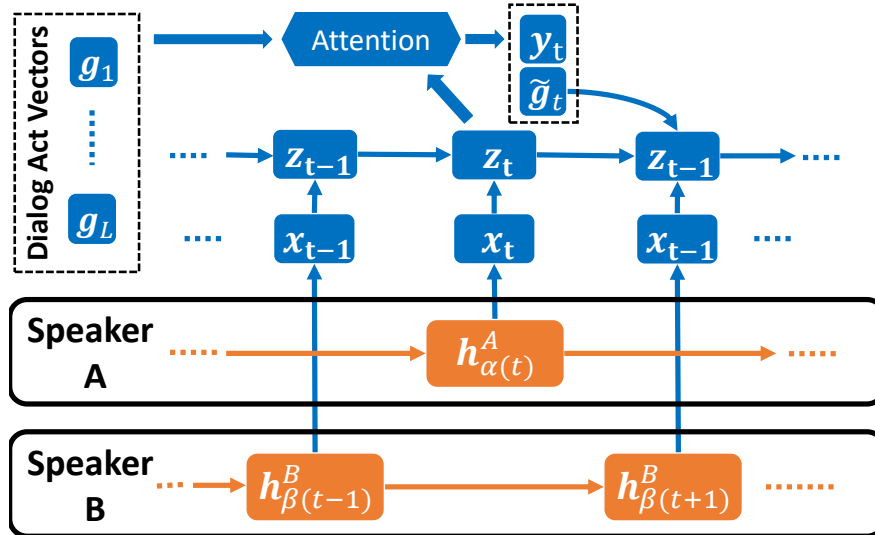


Figure 5.2: The attention-based LSTM tagging model for dialog act classification. The figure only shows the attention operation for turn t . The lower two boxes represent two speaker state trackers.

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., $\mathbf{x}_t = I(\mathbf{h}_{\alpha(t)}^A, \mathbf{h}_{\beta(t)}^B)$ where $I(\cdot, \cdot)$ is a switcher that chooses $\mathbf{h}_{\alpha(t)}^A$ or $\mathbf{h}_{\beta(t)}^B$ 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 $\mathbf{g}_1, \dots, \mathbf{g}_L$. The dialog act probabilities $\mathbf{y}_t \in \mathbb{R}^L$ at turn t are computed using the dot-product attention mechanism, i.e., $\mathbf{y}_t = f(\mathbf{z}_t, [\mathbf{g}_1, \dots, \mathbf{g}_L])$, where $f(\cdot, \cdot)$ is defined as in (5.1), and \mathbf{z}_t is the hidden state vector of the LSTM.

The tagging LSTM computes hidden states as

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

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 §5.2.2. 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.

5.3.3 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 times in the Fisher corpus.

We compare our results to best published results. In [122], 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. [123] propose a discourse-aware RNN LM by treating the dialog act as a conditional variable to the LM. [118–120] focus on building hierarchies of RNNs to model the dialog context using previous utterances or dialog act predictions. Results from [124] and [125] are not directly comparable due to different experiment settings.

Experiment results are summarized in Table 5.8. Our pre-train setting performs on par with previous state-of-the-art supervised models except [123]. Fine-tuning significantly improves the per-

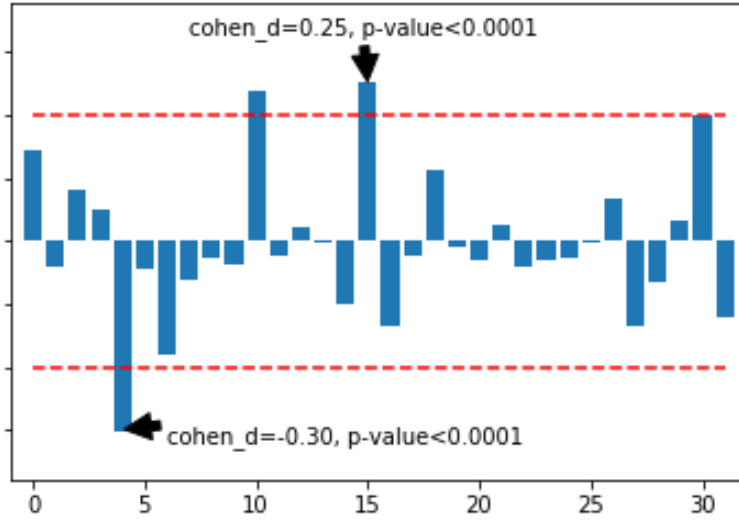
| Model | Acc (%) |
|--|----------------|
| (Kalchbrenner and Blunsom, 2013) [122] | 73.9 |
| (Tran et al., 2017a) [119] | 74.2 |
| (Tran et al., 2017b) [118] | 74.5 |
| (Tran et al., 2017c) [120] | 75.6 |
| (Ji et al., 2016) [123] | 77.0 |
| pre-train | 75.6 |
| pre-train + fine-tune | 77.2 |
| pre-train w/ Fisher + fine-tune | 78.6* |

Table 5.8: 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 ($p < .001$).

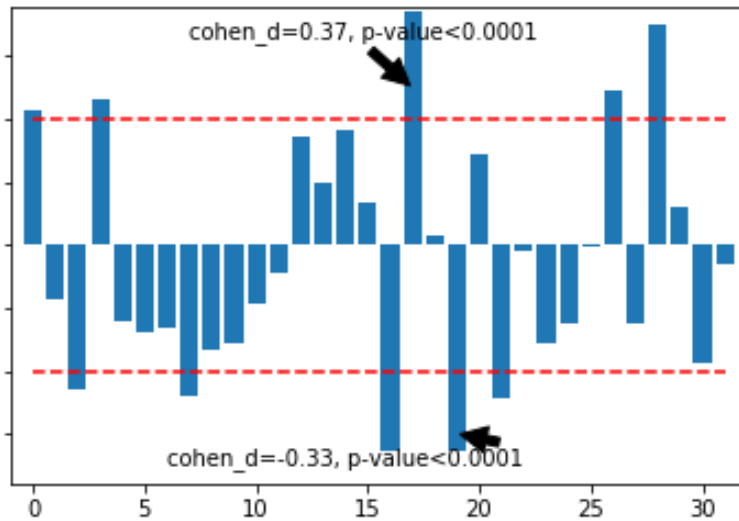
formance and allows the model to achieve a similar accuracy as [123]. The best result is achieved by pre-training the dynamic speaker model with both SwDA and Fisher datasets. The improvement of pre-train w/ Fisher + fine-tune is statistically significant over pre-train + fine-tune based on McNemar’s 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.

5.3.4 Qualitative Analysis

We analyze the latent modes learned on SwDA using the same approach as in §5.2.5. Specific examples are shown in Tables 5.9–5.10. Overall, 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). Many modes characterize statements, probably due to their high relative frequency in the corpus. 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.



(a) same-gender conversations



(b) cross-gender conversations

Figure 5.3: 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.

In addition, we use the speaker gender information⁴ 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 shown in Fig. 5.3. The Cohen-d score [126] 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. 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 [127]. Thus, we carry out the group mean tests on the following two sets: 1) same-gender conversations, and 2) cross-gender conversations. Comparing the same-gender and cross-gender association scores, we observe a difference in the use of the modes for both men and women.

5.4 Summary

In this chapter, we address the problem of modeling speakers from their language using our proposed unsupervised approach. A dynamic speaker model is designed to learn speaker embeddings that are updated as the conversation evolves. The model achieves promising results on two representative tasks in two-party open-domain dialogues: 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.

⁴The Switchboard data was collected in 1990, and only binary gender distributions were recorded. The number of people who chose not to respond to that question was small.

| | | |
|------------|---------|---|
| Statements | Mode-2 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • 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 |

Table 5.9: Utterances for each mode in SwDA dataset: statements.

| | | |
|----------------------|---------|---|
| Backchannel | Mode-0 | <ul style="list-style-type: none"> • yes • yes NONVERBAL |
| | Mode-15 | <ul style="list-style-type: none"> • see • probably • like |
| | Mode-17 | <ul style="list-style-type: none"> • uh • um |
| | Mode-18 | <ul style="list-style-type: none"> • oh oh yeah • oh well • oh okay |
| | Mode-28 | <ul style="list-style-type: none"> • uh huh NONVERBAL • uh huh NONVERBAL NONVERBAL • uh huh ery faint |
| Agreement | Mode-12 | <ul style="list-style-type: none"> • exactly |
| | Mode-20 | <ul style="list-style-type: none"> • yep ause • definitely • absolutely • i agree |
| Quesetion | Mode-8 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • bye • bye bye • appreciation talking to you |

Table 5.10: Utterances for each mode in SwDA dataset: other dialog acts.

Chapter 6

TASK-ORIENTED DIALOGUES

Due to recent interest in conversational assistants, there is an increasing body of research on dialogue systems that assist users in completing tasks, also known as task-oriented dialogue systems. Different from open-domain social-chat dialogue systems, task-oriented dialogue systems are designed for completing pre-defined tasks. A key component of a typical task-oriented dialogue system is the dialogue state tracker, which maintains the task-related information in the conversation using a semantic representation. In this chapter, we develop a new dialogue state tracker based on the proposed dynamic speaker model from Chapter 5.

6.1 Background

For task-oriented dialogue systems, there is usually a given *ontology* that defines all the knowledge information a system can use to understand user goals and complete the target tasks. Specifically, the ontology can define a collection of *slots*, each of which has a type attribute (e.g., `price`) and a value attribute (e.g., `moderate`). For a task-oriented dialogue system, the ontology is usually domain-dependent.

A dialogue state tracker can model the dialogue state at each turn as a set of slots in the ontology. An example of restaurant-booking conversation is illustrated in Figure 6.1. In general, the agent focuses on understanding user goals by extracting slots from each user turn and records the task-related information in the dialogue state. In other words, the dialogue state at turn t contains all the task-related information that an agent has collected from the conversation so far. As shown in the given example, the user is interested in finding a moderately priced Chinese restaurant in the east side. Specifically, the user’s goal is incrementally constructed via a sequence of *turn labels* as shown in the middle of Figure 6.1. The turn label of the first turn consists of a `food` slot with a slot

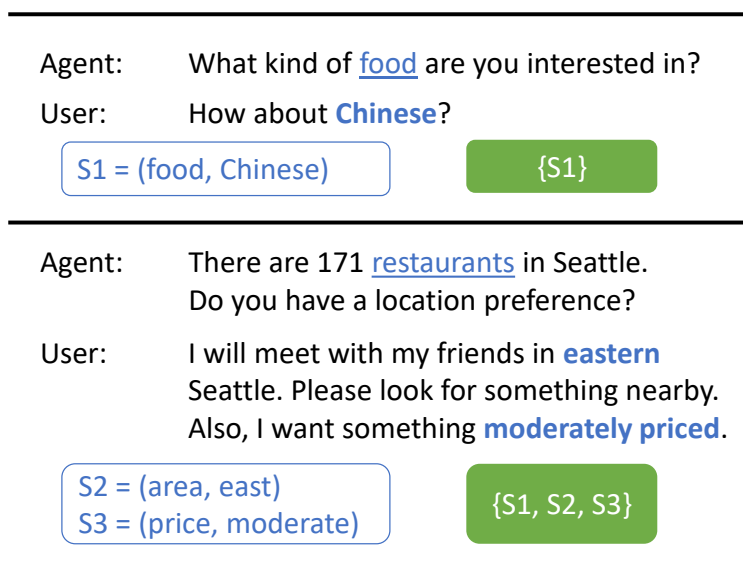


Figure 6.1: A dialogue state tracking example. In this example, a user is looking for a moderately priced Chinese restaurant in the east side. Turn labels are shown in blue boxes, and dialogue states are shown in green boxes.

value `Chinese`, and the turn label of the second turn consists of an `area` slot and a `price` slot with values `east` and `moderate`, respectively. These turn labels are accumulated as the dialogue state at each turn, shown in green boxes on the right side of Figure 6.1.

There are two main challenges for the dialogue state tracking problem. The first challenge comes from rare slots, including both the slot type and slot value. Particularly, the problem of unseen slot values is pretty frequent even under the single-domain setting. To address this problem, recent work has focused on using neural text representation built upon pretrained word embeddings [87, 88]. These embeddings allow the system to leverage the semantic similarity between user utterances and slot labels. Another challenge is to scale the dialogue state tracker to handle tasks spanning multiple domains with more complex and semantically-rich dialogues. In addition to the problem of rare slots, the multi-domain dialogue state tracker is required to perform context switching over a much larger ontology or potentially an open ontology. Existing approaches try to address this issue by enabling the knowledge sharing through the supervised task of domain

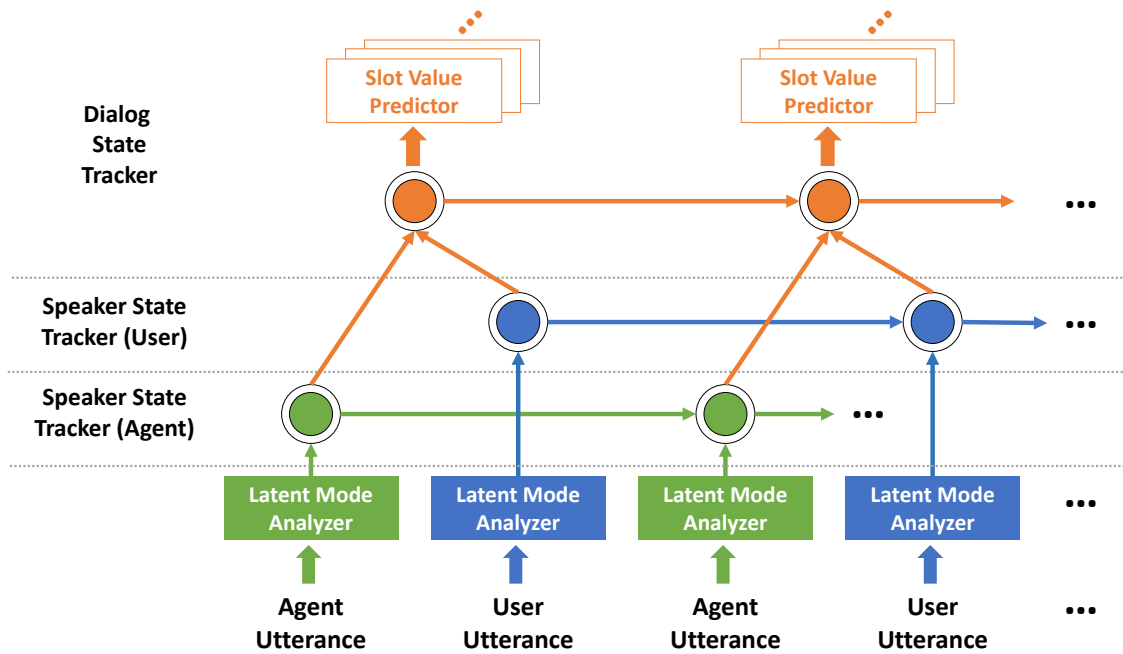


Figure 6.2: Dialogue state tracking using two dynamic speaker models for agent and user, respectively. The dialogue state tracker makes slot value predictions using output from both models.

tracking [128, 129]. Based on this domain detection, the dialogue state tracker then carries out context-aware prediction.

Instead of only relying on supervised knowledge sharing approaches, in this chapter, we propose to leverage the dynamic speaker model proposed in Chapter 5 for knowledge sharing. Specifically, through the unsupervised learning, the resulting embeddings can encode both the slot semantics within and across domains and the domain-switching across dialogues.

6.2 Dialogue State Tracking Model

In this section, we formally describe the model used for the dialog state tracking. Figure 6.2 illustrates the full model. Two dynamic speaker models are used for the agent and the user, respectively. They are trained jointly and share model parameters, although the sharing is optional. The dialogue state tracker uses output from both models and predicts individual slot values in the belief state.

The latent model analyzer and the speaker state tracker are almost identical to the dynamic speaker model proposed in Chapter 5, with some changes on the utterance encoder.

6.2.1 Changes on the Utterance Encoder of the Dynamic Speaker Model

Different from Chapter 5 where only pre-trained word embeddings from [110] are used, here we additionally use a convolutional neural network (CNN) to construct word vectors from their corresponding characters to handle unseen words in slot types and slot values. Following [130], each word $w_{t,n}$ ($n = 1, \dots, N$) in the current utterance is encoded by a CNN into a character-based word embedding $\mathbf{c}_{t,n} \in \mathbb{R}^d$, *i.e.* $\mathbf{c}_{t,n} = \text{CNN}(w_{t,n})$. $\mathbf{c}_{t,n}$ is concatenated with the pre-trained word embedding $\mathbf{w}_{t,n} \in \mathbb{R}^d$ as the final word embedding used by the dynamic speaker models.

Recall that the latent mode analyzer in Chapter 5 uses a bi-directional LSTM (bi-LSTM) to encode the utterance at turn t into a sequence of hidden states, or contextualized word embeddings $\mathbf{e}_{t,n} = [\mathbf{e}_{t,n}^F, \mathbf{e}_{t,n}^B]$, $n = 1, \dots, N$, and a summary vector $\mathbf{v}_t = [\mathbf{e}_{t,1}^B, \mathbf{e}_{t,N}^F]$, where $\mathbf{e}^F, \mathbf{e}^B$ come from the forward and backward LSTMs, respectively. The summary vector is further used for computing a local user mode vector $\tilde{\mathbf{u}}_t$ from a set of K global user mode vectors as described in Chapter 5.

The speaker state tracker then incorporates (in the same way as in Chapter 5) the current local user mode information in $\tilde{\mathbf{u}}_t$ with all previous extracted local user modes through another single directional LSTM and produces a speaker state vector \mathbf{h}_t at turn t .

6.2.2 Dialogue State Tracker

In order to fully utilize the semantic similarity between turn labels in the ontology and the corresponding lexical variations in natural language utterances, various word-level encoding of slot labels are adopted in previous work [87, 88, 128]. Here, we re-use the character-level CNN and word-level bi-LSTM in the latent mode analyzer to encode individual slot types and values, so that the semantic information is shared between utterances and slots. Given the m -th slot type in the ontology with words, l_1^m, \dots, l_J^m , a sequence of hidden states $\mathbf{q}_1^m, \dots, \mathbf{q}_J^m$ are obtained from the bi-LSTM. Using these hidden states, a self-attention model is applied to compute the slot type

vector as

$$\mathbf{o}_m = \sum_j \sum_i a^s(\mathbf{q}_i^m, \mathbf{q}_j^m) \mathbf{q}_i^m,$$

where $a(\cdot, \cdot)$ computes the normalized attention score. Here, the attention score is computed using a linear dot-product as in [30]. The slot value vectors are computed in the same way using the same parameters.

After that, for turn t , each slot type vector \mathbf{o}_m is used to attend over all the contextualized word embeddings $\mathbf{e}_{t,n}$, $n = 1, \dots, N$, to search for most relevant information expressed and the relevant information is summarized as

$$\tilde{\mathbf{e}}_{t,m} = \sum_n b(\mathbf{e}_{t,n}, \mathbf{o}_m) \mathbf{e}_{t,n},$$

where $b(\cdot, \cdot)$ computes the normalized attention score between the n -th word and the m -th slot type. Here, the contextualized word embeddings comprise the words in the user utterance at turn t and its preceding agent utterance.

In the single-domain setting, the model predicts the slot value for the m -th slot type using the contextualized word embeddings $\tilde{\mathbf{e}}_{t,m}$, the speaker state vector for the user turn \mathbf{h}_t^u , and the speaker state vector for the preceding agent turn \mathbf{h}_{t-1}^a . The predictions for individual slot types are carried out independently, and the predicted probability of the slot value $y_{t,m}^v$ is

$$\mathbb{P}(y_{t,m}^v) \propto \exp(\mathbf{W}_m \tilde{\mathbf{e}}_{t,m} + \mathbf{U}_m [\mathbf{h}_{t-1}^a, \mathbf{h}_t^u]),$$

where $\mathbf{W}_m \in \mathbb{R}^{|\mathcal{Y}_m| \times d}$ and $\mathbf{U}_m \in \mathbb{R}^{|\mathcal{Y}_m| \times d}$ are weight matrices, and $|\mathcal{Y}_m|$ is the number of possible slot values for m -th slot type. For the multi-domain setting, the slot value vectors are used as additional input for prediction.

6.2.3 Model Training and Inference

We pretrain the dynamic speaker model without using the turn labels. Then, the full dialogue state tracking model is trained to predict turn labels. We follow the same training protocol as in Chapter 5.

| | Single-WoZ | Multi-WoZ |
|---------------|-------------------|------------------|
| # Dialogues | 600 | 8438 |
| # Turns | 4472 | 113556 |
| Avg. # turns | 7.5 | 13.5 |
| # Slot Types | 4 | 24 |
| # Slot Values | 99 | 4510 |
| # Domains | 1 | 6 |

Table 6.1: Data statistics of the Single-WoZ dataset and the Multi-WoZ dataset.

At inference time, we incrementally construct the dialogue state turn by turn using the predicted turn labels. Specifically, slots are either added to the dialogue state or updated with the newer value. Following Zhong et al. [88], slots are not removed from the dialogue state once added.

6.3 Experiment

6.3.1 Data

To evaluate our model, we use two public dialogue state tracking datasets collected under the Wizard-of-Oz (WoZ) setting, i.e., the single-domain WoZ (Single-WoZ) corpus with dialogues for restaurant reservation [84] and the multi-domain WoZ (Multi-WoZ) corpus with dialogues spanning several domains [131]. In the Multi-WoZ corpus, a single conversation can involve bookings on multiple domains, including restaurant, hotel, attraction, taxi, train, hospital, and police. Table 6.1 shows the statistics for these two datasets. For both datasets, we use joint goal accuracy as the evaluation metric, which is the turn-level accuracy that counts the exact match of the predicted dialogue state against the gold dialogue state at each turn.

6.3.2 Experiment with Single-WoZ

For the Single-WoZ dataset, we include the two recent state-of-the-art (SOTA) models for comparison:

- neural belief tracker (NBT) [87], which leverages CNNs over word embeddings for sharing

| Model | Joint Goal Accuracy (%) |
|------------------|--------------------------------|
| GLAD [88] | 88.1 |
| NBT [87] | 84.4 |
| Ours | 90.3 |
| w/o pretraining | 87.5 |
| w/o latent modes | 88.8 |

Table 6.2: Joint goal accuracy on the Single-WoZ test set.

the semantics between slot labels and utterances,

- globally-locally self-attentive dialogue state tracker (GLAD) [88], which uses both slot-dependent and global networks which enables the sharing among related slots.

Different from these two models which use extra dialogue features such as system actions, our model only uses agent and user utterances as input. We hypothesize that some dialogue features may be captured in the pretraining process.

We further include two variants of our proposed model for comparison:

- w/o pretraining, which skips pretraining the dynamic speaker model, but it still uses pre-trained word embeddings,
- w/o latent modes, which excludes the latent modes from the input to the speaker state tracker.

For the second variant, the dynamic speaker model is still pretrained.

The experiment results are summarized in Table 6.2. Our full model outperforms GLAD and NBT by 2.2 and 5.9 points, respectively. While both GLAD and NBT use pretrained word embeddings to share semantic information between slots and utterances, our pretraining additionally captures dialogue-level information, bringing 2.8 points improvement over the “w/o pretrain” variant. Comparing with the variant “w/o latent modes”, our full model achieves 1.5 point improvement, suggesting the latent modes may have captured some useful information that benefit the dialogue state tracking.

| Model | Joint Goal Accuracy (%) |
|-------------------------------|--------------------------------|
| HyST [132] | 38.1 |
| TRADE [129] | 45.6 |
| Ours | 40.2 |
| w/o pretraining | 37.5 |
| w/o latent modes | 38.0 |
| w/ dialogue state fine-tuning | 46.1 |

Table 6.3: Joint goal accuracy on the Multi-WoZ test set.

6.3.3 Experiment with Multi-WoZ

In this subsection, we evaluate our model on the multi-domain dialogue state tracking. In addition to our full model and the two variants (w/o pretraining, w/o latent modes) described in Subsection 6.3.2, the following two SOTA models on Multi-WoZ are included for comparison:

- HyST [132], which is built upon a hierarchical encoder model that combines a closed-vocabulary discriminative sub-model and an open-vocabulary generative sub-model with copy mechanism,
- TRADE [129], which is a generative model based on an encoder-decoder network with copy mechanism.

Different from these two models, our full model is a closed-vocabulary discriminative model. While we also use a hierarchical encoder, it is built upon two dynamic speaker models sharing the same parameters instead. Note that unlike other models which construct the dialogue state turn by turn using predicted turn labels and thus suffer from error propagation, TRADE directly predicts the dialogue state at each turn independently, which seems to be more effective on the dialogue state tracking task. Therefore, we further include a model variant (w/ dialog state fine-tuning) which fine-tunes the full model by directly predicting the dialogue state.

The experiment results on Multi-WoZ are summarized in Table 6.3. Comparing the models that predict turn labels, our model outperforms HyST by 2.1 points. Consistent with results in Subsection 6.3.2, we observe both pretraining and latent modes are useful for dialogue state track-

ing. It can be observed that using the dialogue state fine-tuning significantly improves the model performance by 5.9 points, surpassing TRADE. Besides reducing the mismatch between training and inference, we hypothesize that by directly predicting dialogue states, the model is enforced to learn to make full use of the dialogue context information and thus effectively share information among different domains.

Lastly, we explore the use of BERT [9] to obtain contextualized word representations, which has recently achieved impressive results across many NLP tasks. Specifically, we use BERT to replace the bi-LSTM in the dynamic speaker model for encoding utterances and slot labels. Due to the huge memory cost for backpropagating errors, we use a fixed BERT-base model for encoding slots labels, and another pretrained BERT-base model with fine-tuning for encoding utterances. It turns out the resulting model has low accuracy compared with other models. We hypothesize that this might be caused by optimizing difficulties when building hierarchical models upon BERT.

6.4 Summary

In this chapter, we apply the dynamic speaker model proposed in Chapter 5 for task-oriented dialogues. Specifically, we develop a dialogue state tracking model based on the unsupervised dynamic speaker model. In addition to leveraging existing neural approaches for knowledge sharing through utterances and labels, our approach aims at encoding the way that multiple-domain bookings are carried out through unsupervised language prediction. Based on the experiments on two recent dialogue state tracking datasets, our proposed approach achieves improvement over recent state-of-the-art models, demonstrating the potential effectiveness of the unsupervised dynamic speaker model for task-oriented dialogues.

Chapter 7

CONCLUSION

To conclude, we first provide a summary of the work carried out in this thesis in Section 7.1. After that, impacts of this thesis are discussed in Section 7.2. Finally, we discuss future directions for work with contextualized representations for interactive language in Section 7.3.

7.1 Summary

Given that representing language in context is the key to improving NLP, the main focus of this thesis is to develop unsupervised contextualized neural text representations for interactive language. As prior work has focused on monologues, our work is a pioneer in addressing interactive language. Specifically, this thesis focuses on leveraging two types of context: local text context consisting of neighboring words or sentences and global mode context referring to characteristics of the interlocutor, social context and community identity that impact language use. Different from recent extensive research effort on developing contextualized text representations for written monologues which mainly concerns local text context, the text representation developed in this thesis additionally makes use of the structure of interactive language, *i.e.* comment-reply pairs on online discussions and segmented dialogue turns associated with alternating speakers.

In Chapter 3, we introduced a new unsupervised learning framework that provides a factored representation of interactive language. Specifically, the framework is built upon the encoder-predictor model and the latent dynamic context model. By providing a unifying view of multiple popular sequence models, the encoder-predictor framework was leveraged to compose a factored representation corresponding to various types of context including local text context and global mode context. The latent dynamic context model updates the global context information represented in discrete latent mode vectors in an online fashion. By applying the latent dynamic context

model jointly with the encoder-predictor model, the resulting embeddings can provide some interpretation on what is captured.

In Chapter 4, based on our proposed framework, we developed a factored neural model for multi-party discussions from a large collection of Reddit comments. Through evaluation on the task of predicting the community endorsement for three online communities differing in topic trends and writing styles, we showed that our approach consistently outperforms multiple document embedding baselines. Analyses of the learned discrete modes showed community-related style and topic characteristics are captured.

In Chapter 5, for dialogue application where participants contribute multiple turns, we extended our framework to learning dynamically updated speaker embeddings. We used the learned dynamic speaker embeddings in two representative tasks in social-chat dialogues: topic decision prediction in socialbot dialogues and dialogue act prediction in human-human conversations. Empirical results showed that using the dynamic speaker embeddings significantly outperforms the baselines in both tasks.

In Chapter 6, we applied the dynamic speaker embedding model to two-party task-oriented dialogues. We developed a new dialogue state tracking model to leverage the knowledge encoded in the unsupervised dynamic speaker model about the way multiple domain tasks are carried out. In experiments on two recent dialogue state tracking datasets, our approach achieved improvement over state-of-the-art models, demonstrating the potential effectiveness of the unsupervised dynamic speaker model for task-oriented dialogues.

7.2 Impacts and Implications

As representing language with context is essential for NLP tasks, there is growing interest in developing better contextualized text representations. This thesis represents early work on developing unsupervised contextualized representations for interactive language.

- Our approach of using a factored representation where subvectors are linked to different contexts by training with multiple objectives can be a useful way of joint learning from multiple structured contexts, *i.e.* local text context and global mode context in our case.

Specifically, the predictive model developed in this thesis has been shown to benefit from the structured context-aware text representation across multiple tasks under the interactive language setting. Moreover, the factored approach can potentially be useful for developing contextualized representations for monologues.

- The study of three representative interaction scenarios with different characteristics demonstrates the generalizability of our unsupervised neural representation framework. It can potentially benefit the design of variants of unsupervised text representation models for a large body of other types of interactive language, such as multi-party meetings, customer service call center, political debates and so on.
- An important focus of this thesis was to develop interpretable neural representations. Our approach of discovering latent interpretable modes would be of interest to other applications where model interpretability is often desirable and sometimes critical. We showed that the interpretation can be helpful for analyzing human interaction strategies or styles which can be used as a tool for developing dialogue systems. In addition, it could be used to identify characteristics of community language, such as regional language, social status, political ideology, *etc.*

7.3 *Future Directions*

In this thesis, we have made the first step in developing unsupervised contextualized text representations for interactive language. There is great potential for the factored representation to be useful for modeling multiple contexts beyond the local text context and global mode context considered in this thesis.

Knowledge-aware representations: One interesting extension of our text representation is to learn jointly with external knowledge. One type of knowledge is in the form of entity-entity relations encoded in knowledge bases, which have been heavily leveraged by NLP tasks such as question answering and semantic parsing. We think this form of external knowledge can also be useful for tasks with interactive language. Take the topic recommendation problem in social bot dialogues, for example. If the user shows interest in the movie featured by a specific actor, with the knowledge

of other movies played by the same actor, the representation can be useful for recommending more intriguing follow-up topics to be discussed. Specifically, instead of replying solely on raw text, an entity linking model can be leveraged to associate those entities mentioned in dialogues with their corresponding sub-graphs including edges connecting to other entities. By adding a subvector used to predict the edges and/or the neighboring entities, the factored representation approach can be extended to incorporating knowledge from knowledge bases.

Universal embeddings for multiple interactive language settings: In this thesis, based on the proposed general unsupervised text representation framework, we developed two different variants for multi-party discussion and two-party dialogue respectively. It would be of interest to pretrain one universal architecture for multiple interactive language settings which can potentially benefit from learning from a large amount of data. One plausible approach would be to leverage contextualized text representations pretrained in a similar fashion as monologues, such as ELMO and BERT. Then, an additional unsupervised model grounded on the specific interactive context can be applied for learning shared representations for settings that share similar interactive structure. For example, the encoder used by the latent model analyzer of the dynamic speaker model studied in Chapter 5 can be replaced with the BERT model. By leveraging the segmented turns from the user side, the resulting representation can be useful for both social-chat dialogues and task-oriented dialogues by encoding speaker state information.

In addition, the learned embeddings seem to capture community or speaker characteristics based on unsupervised learning from their language use. Therefore, it might be interesting to explore using embeddings learned through unsupervised learning for developing context-aware NLP models.

Personalized recommendation and language understanding: One promising direction would be using the unsupervised text representation to serve as the user/community identity information for personalized content recommendation and language understanding. In this thesis, we have shown promising results for using the embeddings which are learned from the language in an unsupervised fashion for post recommendation for online discussion and topic recommendation for social-chat dialogue. It might be interesting to explore scientific paper recommendation by learning scientific

community characteristics using unsupervised text representation approach. Moreover, because many language understanding tasks can potentially benefit from speaker identity information, it would be very interesting to explore using the dynamically derived speaker embeddings for sentiment and emotion analysis.

Context-aware and disentangled latent language generation: In [35], the authors have demonstrated the use of persona information to bias the neural language generation model to mimic the speaking style of the corresponding speaker. Given the persona information is not generally available, it would be a potentially interesting direction to explore using the unsupervised text representation approach to derive user embeddings for neural language generation models. Moreover, the factored representation approach discussed in this thesis might be effective for deriving disentangled latent representations. For example, we can divide the latent vector into sub-vectors linked with separate training objectives, each of which can be a specific factor for controllable text generation.

BIBLIOGRAPHY

- [1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proc. Workshop at Int. Conf. Learning Representations*, 2013.
- [2] Tomas Mikolov, Wen-Tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 746–751, 2013.
- [3] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2014.
- [4] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 3276–3284, 2015.
- [5] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proc. Int. Conf. Machine Learning (ICML)*, pages 3104–3112, 2014.
- [6] Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, , and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*, 2015.
- [7] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, 2017.
- [8] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*, 2018.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, 2019.

- [10] Tomas Mikolov and Geoffrey Zweig. Context dependent recurrent neural network language model. In *Proc. IEEE Spoken Language Technologies Workshop*, 2012.
- [11] Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. Sparse additive generative models of text. In *Proc. Int. Conf. Machine Learning (ICML)*, 2011.
- [12] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *J. Machine Learning Research*, pages 1137–1155, 2003.
- [13] Andriy Mnih and Geoffrey Hinton. Three new graphical models for statistical language modelling. In *Proc. Int. Conf. Machine Learning (ICML)*, 2007.
- [14] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 2265–2273, 2013.
- [15] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 302–308, 2014.
- [16] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, pages 1045–1048, Makuhari, Japan, 2010.
- [17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 5998–6008, 2017.
- [18] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proc Int. Conf. Information and Knowledge Management (CIKM)*, 2013.
- [19] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Gregoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proc Int. Conf. Information and Knowledge Management (CIKM)*, 2014.
- [20] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. Int. Conf. Machine Learning (ICML)*, 2008.
- [21] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 655–665, 2014.

- [22] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 3104–3112, 2014.
- [23] Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proc. Deep Learning and Unsupervised Feature Learning Workshop at NIPS*, 2010.
- [24] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 1556–1566, 2015.
- [25] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: encoder-decoder approaches. In *Proc. Eighth Workshop Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, pages 103–111, 2014.
- [26] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.
- [27] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahadanau, Fethhi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1724–1734, 2014.
- [28] Richard Socher, Cliff C. Lin, Andrew Y. Ng, and Christopher D. Manning. Parsing natural scenes and natural language with recursive neural networks. In *Proc. Int. Conf. Machine Learning (ICML)*, 2011.
- [29] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.
- [30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, 2017.
- [31] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for sentence summarization. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 379–389, 2015.

- [32] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 2755–2763, 2015.
- [33] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: a neural image caption generator. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.
- [34] Oriol Vinyals and Quoc Le. A neural conversation model. In *Proc. ICML Deep Learning Workshop*, 2015.
- [35] Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. A persona-based neural conversation model. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, 2016.
- [36] Pei-Hao Su, David Vandyke, Milica Gasic, Nikola Mrksic, Tsung-Hsien Wen, , and Steve Young. Reward shaping with recurrent neural networks for speeding up on-line policy learning in spoken dialogue systems. In *Proc. SIGdial Workshop Discourse and Dialogue*, 2015.
- [37] Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. On-line active reward learning for policy optimisation in spoken dialogue systems. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, 2016.
- [38] John Langshaw Austin. How to do things with words. In *William James lectures*. Cambridge: Harward University Press, 1962.
- [39] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.
- [40] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 2431–2439, 2015.
- [41] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2017.
- [42] Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional LSTM feature representation. *Trans. Assoc. for Computational Linguistics (TAACL)*, 4:313–327, 2016.

- [43] Hao Cheng, Hao Fang, Xiaodong He, Jianfeng Gao, and Li Deng. Bi-directional attention with agreement for dependency parsing. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2016.
- [44] Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. Dependency parsing as head selection. In *Proc. European Chapter Assoc. for Computational Linguistics (EACL)*, 2017.
- [45] Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2017.
- [46] Svitlana Volkova, Theresa Wilson, and David Yarowsky. Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1815–1827, 2013.
- [47] Bongwon Suh, Lichan Hong, Peter Pirolli, and Ed H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in Twitter network. In *Proc. IEEE Second Intern. Conf. Social Computing*, 2010.
- [48] Chenhao Tan, Lillian Lee, and Bo Pang. The effect of wording on message propagation: Topic- and author-controlled natural experiments on Twitter. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, 2014.
- [49] Marco Guerini, Carlo Strapparava, and Gozde Ozba. Exploring text virality in social networks. In *Proc. Int. AAI Conf. Web and Social Media (ICWSM)*, 2011.
- [50] Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proc. WWW*, 2016.
- [51] Tim Althoff, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. How to ask for a favor: A case study of the success of altruistic request. In *Proc. Int. AAI Conf. Web and Social Media (ICWSM)*, 2014.
- [52] Zhongyu Wei, Yang Liu, and Yi Li. Ranking argumentative comments in the online forum. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 195–200, 2016.
- [53] Aaron Jaech, Vicky Zayats, Hao Fang, Mari Ostendorf, and Hannaneh Hajishirzi. Talking to the crowd: What do people react to in online discussions? In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2026–2031, 2015.
- [54] Hao Fang, Hao Cheng, and Mari Ostendorf. Learning latent local conversation modes for predicting community endorsement in online discussions. In *Proc. Int. Workshop Natural Language Process. for Social Media*, 2016.

- [55] Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Proc. SIGIR*, 1999.
- [56] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *Proc. Int. Conf. Computational Linguistics (COLING)*, 2010.
- [57] Lu Wang, Hema Raghavan, Claire Cardie, and Vittorio Castelli. Query-Focused Opinion Summarization for User-Generated Content. In *Proc. Int. Conf. Computational Linguistics (COLING)*, pages 1660–1669, 2014.
- [58] Lu Wang, Larry Heck, and Dilek Hakkani-Tur. Leveraging semantic web search and browse sessions for multi-turn spoken dialog systems. In *Proc. Int. Conf. Acoustic, Speech, and Signal Process. (ICASSP)*, pages 4082–4086, 2014.
- [59] 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 Pettigrew. Conversational AI: The science behind the alexa prize. In *Proc. Alexa Prize 2017*, 2017.
- [60] Liangjie Hong, Ovidiu Dan, and Brian D. Davison. Predicting popular messages in Twitter. In *Proc. WWW*, 2011.
- [61] Himabindu Lakkaraju, Julian McAuley, and Jure Leskovec. What’s in a name? Understanding the interplay between titles, content, and communities in social media. In *Proc. Int. AAAI Conf. Web and Social Media (ICWSM)*, 2013.
- [62] Ingrid Zukerman and Diane Litman. Natural language processing and user modeling. *User Modeling and User-Adapted Interaction*, 11:129–158, 2001.
- [63] Elaine Rich. User modeling via stereotypes. *Cognitive Science*, 3:329–354, 1979.
- [64] David N. Chin. User modeling in UC, the UNIX consultant. In *Proc. Computer Human Interactions (CHI)*, pages 24–28, 1986.
- [65] Derek Sleeman. UMFE: A user modelling front-end subsystem. *Int. J. Man-Machine Studies*, 23:71–88, 1985.
- [66] Cécile L. Paris. *The Use of Explicit User Models in a Generation System for Tailoring Answers to the User’s Level of Expertise*. PhD thesis, Columbia University, 1987.

- [67] Eduard Hovy. Generating natural language under pragmatic constraints. *Journal of Pragmatics*, 11:689–710, 1987.
- [68] James F. Allen and C. Raymond Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178, 1980.
- [69] Sandra Carberry. Tracking user goals in an information-seeking environment. In *Proc. AAAI Conf. Artificial Intelligence*, pages 59–63, 1983.
- [70] Diane J. Litman. Linguistic coherence: a plan-based alternative. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 215–223, 1986.
- [71] Johanna D. Moore and Cecile Paris. Exploiting user feedback to compensate for the unreliability of user models. *User Modeling and User-Adapted Interaction*, 2:287–330, 1992.
- [72] François Mairesse and Marilyn Walker. Automatic recognition of personality in conversation. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*, pages 85–88, 2006.
- [73] David DeVault, Ron Artstein, Grace Benn, Teresa Dey, Ed Fast, Alesia Gainer, Kallirroi Georgila, Jon Gratch, Arno Hartholt, Margaux Lhommet, et al. SimSensei kiosk: A virtual human interviewer for healthcare decision support. In *Proc. Int. Conf. Autonomous Agents and Multi-agent Systems*, pages 1061–1068, 2014.
- [74] Pascale Fung, Anik Dey, Farhad Bin Siddique, Ruixi Lin, Yang Yang, Yan Wan, and Ricky Ho Yin Chan. Zara the supergirl: An empathetic personality recognition system. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL) (System Demonstrations)*, 2016.
- [75] Hao Fang, Hao Cheng, Elizabeth Clark, Ariel Holtzman, Maarten Sap, Mari Ostendorf, Yejin Choi, and Noah Smith. Sounding Board – University of Washington’s Alexa Prize submission. In *Proc. Alexa Prize*, 2017.
- [76] Ali Ahmadvand, Ingyu Choi, Harshita Sahijwani, Justus Schmidt, Mingyang Sun, Sergey Volokhin, Zihao Wang, and Eugene Agichtein. Emory IrisBot: An open-domain conversational bot for personalized information access. In *Proc. Alexa Prize 2018*, 2018.
- [77] Daniel Preoțiuc-Pietro, Wei Xu, and Lyle Ungar. Discovering user attribute stylistic differences via paraphrasing. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3030–3037, 2016.

- [78] Anders Johannsen, Dirk Hovy, and Anders Søgaard. Cross-lingual syntactic variation over age and gender. In *Proc. Conf. Computational Natural Language Learning (CoNLL)*, pages 103–112, 2015.
- [79] Dirk Hovy and Anders Søgaard. Tagging performance correlates with author age. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 483–488, 2015.
- [80] Shachar Mirkin, Scott Nowson, Caroline Brun, and Julien Perez. Motivating personality-aware machine translation. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1102–1108, 2015.
- [81] Veronica Lynn, Youngseo Son, Vivek Kulkarni, Niranjan Balasubramanian, and H. Andrew Schwartz. Human centered nlp with user-factor adaptation. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1146–1155, 2017.
- [82] Steve Young. Talking to machines (statistically speaking). In *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2002.
- [83] Jason D. Williams, Antoine Raux, and Matthew Henderson. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33, 2016.
- [84] Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M. Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. In *Proc. European Chapter Assoc. for Computational Linguistics (EACL)*, pages 438–449, 2017.
- [85] Matthew Henderson, Milica Gašić, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young. Discriminative spoken language understanding using word confusion networks. In *Proc. Spoken Language Technology Workshop (SLT)*, 2012.
- [86] Matthew Henderson, Blaise Thomson, and Steve Young. Word-based dialog state tracking with recurrent neural networks. In *Proc. SIGdial Workshop Discourse and Dialogue*, 2014.
- [87] Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, 2017.
- [88] Victor Zhong, Caiming Xiong, and Richard Socher. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, 2018.

- [89] Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proc. AAAI Conf. Artificial Intelligence*, 2016.
- [90] Ji He, Mari Ostendorf, Xiaodong He, Jiansu Chen, Jianfeng Gao, Lihong Li, and Li Deng. Deep reinforcement learning with a combinatorial action space for predicting popular Reddit threads. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 195–200, 2016.
- [91] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *J. American Society for Information Science*, 41(6):391–407, 1990.
- [92] Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. Sparse overcomplete word vector representations. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, 2015.
- [93] Trang Tran and Mari Ostendorf. Characterizing the language of online communities and its relation to community reception. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1030–1035, 2016.
- [94] Hao Cheng, Hao Fang, and Mari Ostendorf. A factored neural network model for characterizing online discussions in vector space. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2296–2306, 2017.
- [95] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [96] Tomas Mikolov and Geoffrey Zweig. Context dependent recurrent neural network language model. In *Proc. IEEE Spoken Language Technologies Workshop*, 2012.
- [97] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.
- [98] Jason D. M. Rennie and Nathan Srebro. Loss functions for preference levels: regression with discrete ordered labels. In *Proc. Int. Joint Conf. Artificial Intelligence*, 2005.
- [99] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Machine Learning Research*, 3:993–1022, March 2003.

- [100] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *J. Machine Learning Research*, 9, Nov. 2008.
- [101] Kevin K. Bowden, Jiaqi Wu, Wen Cui, Juraj Juraska, Vrindavan Harrison, Brian Schwarzmann, Nick Santer, and Marilyn Walker. SlugBot: Developing a computational model and framework of a novel dialogue genre. In *Proc. Alexa Prize 2018*, 2018.
- [102] Aaron Jaech and Mari Ostendorf. Personalized language model for query auto-completion. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 700–705, 2018.
- [103] Milad Shokouhi. Learning to personalize query auto-completion. In *SIGIR*, pages 103–112. ACM, 2013.
- [104] Deepak Agarwal, Bee-Chung Chen, and Bo Pang. Personalized recommendation of user comments via factor models. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2011.
- [105] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 3111–3119, 2013.
- [106] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*, 2019.
- [107] Hao Cheng, Hao Fang, and Mari Ostendorf. A dynamic speaker model for conversational interactions. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*, pages 2772–2785, 2019.
- [108] Hao Fang, Hao Cheng, Elizabeth Clark, Ariel Holtzman, Maarten Sap, Mari Ostendorf, Yejin Choi, and Noah A. Smith. Sounding board: University of washington’s alexa prize submission. In *Proc. of Alexa Prize*, 2017.
- [109] Hao Fang, Hao Cheng, Maarten Sap, Elizabeth Clark, Ariel Holtzman, Yejin Choi, Noah Smith, and Mari Ostendorf. Sounding Board – a user-centric and content-driven social chatbot. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL) (System Demonstrations)*, 2018.
- [110] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

- [111] Hao Fang. *Building a User-Centric and Content-Driven Socialbot*. PhD thesis, University of Washington, 2019.
- [112] Abhilasha Ravichander and Alan Black. An empirical study of self-disclosure in spoken dialogue systems. In *Proc. SIGdial Meeting Discourse and Dialogue*, pages 253–263, 2018.
- [113] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 3079–3087, 2015.
- [114] John L. Austin. *How To Do Things with Words*. Harvard University Press, Cambridge, MA, 2nd edition, 1975.
- [115] John R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.
- [116] Dan Jurafsky, Elizabeth Shriberg, , and Debra Biasca. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical report, University of Colorado, Boulder, 1997.
- [117] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373, 2000.
- [118] Quan Hung Tran, Ingrid Zukerman, and Gholamreza Haffari. A hierarchical neural model for learning sequences of dialogue acts. In *Proc. European Chapter Assoc. for Computational Linguistics (EACL)*, pages 428–437, 2017.
- [119] Quan Hung Tran, Gholamreza Haffari, and Ingrid Zukerman. A generative attentional neural network model for dialogue act classification. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 524–529, 2017.
- [120] Quan Hung Tran, Ingrid Zukerman, and Gholamreza Haffari. Preserving distributional information in dialogue act classification. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2151–2156, 2017.
- [121] Christopher Cieri, David Graff, Owen Kimball, Dave Miller, and Kevin Walker. Fisher english training speech part 1 transcripts LDC2004T19. Web Download, 2004.
- [122] Nal Kalchbrenner and Phil Blunsom. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126, 2013.

- [123] Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 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, 2016.
- [124] Ji Young Lee and Franck Dernoncourt. Sequential short-text classification with recurrent and convolutional neural networks. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*, pages 515–520, 2016.
- [125] Yang Liu, Kun Han, Zhao Tan, and Yun Lei. Using context information for dialog act classification in dnn framework. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2170–2178, 2017.
- [126] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 1988.
- [127] Constantinos Boulis and Mari Ostendorf. A quantitative analysis of lexical differences between genders in telephone conversations. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 435–442, 2005.
- [128] Osman Ramadan, Paweł Budzianowski, and Milica Gašić. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437, 2018.
- [129] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 808–819, 2019.
- [130] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.
- [131] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026, 2018.
- [132] Rahul Goel, Shachi Paul, and Dilek Hakkani-Tür. Hyst: A hybrid approach for flexible and accurate dialogue state tracking. In *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2019.