MODELING CONTEXTUAL INFORMATION FOR CHIT-CHAT CONVERSATION

LI YANRAN

PhD

The Hong Kong Polytechnic University

2021

The Hong Kong Polytechnic University

Department of Computing

Modeling Contextual Information for Chit-chat Conversation

Li Yanran

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

December 2020

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

_____(Signed)

_____LI Yanran_____(Name of student)

Abstract

The emergence of mobile devices and messaging applications has revolutionized the way that information propagates among individuals, and triggers the demand of virtual conversational agents for assisting and accompanying human users. This presents unprecedented challenges and opportunities and drives many researchers to study how to properly respond to the user based on a given conversation context. In this thesis, we aim at incorporating extra information like knowledge, emotion and intention into open-domain chatbots, which aims to encourage the informativeness and coherence of the generated responses.

In specific, we identify three main research problems to be addressed in open-domain conversation response generation, i.e., *1. How to explore the benefits of extra information for conversation context modeling and response generation when building open-domain chatbots? 2. How to develop an effective chatbot to learn the information change through turns of conversations and consider the dependencies among them? 3. How to understand conversation context better through capturing the interactions between extra information and conversation utterances and improve conversation coherence in a holistic view?*

To address the aforementioned problems, we deploy several approaches based on Seq2Seq models inspired by the recent advances in neural response generation. Because conversations are inherited with discourse structures, we divide the thesis into three parts, where each part concentrates on a certain level of conversation

structure.

In the first part (work 1 and 2), we investigate research problem 1 under the setting of basic level of conversation, i.e., utterance-level. In order to improve utterance-level coherence and alleviate the data sparsity issue, we develop two conditional conversation models to consider knowledge and emotion information, respectively. In work 1, we focus on knowledge incorporation and utilize conversation-related knowledge to generate entity-aware responses. On two movie conversation corpus, the proposed knowledge-grounded chatbot significantly outperforms other four knowledge-grounded models. In work 2, we shift the attention to emotion-incorporation and present a conditional variational model for controlled response generation. The main idea is to introduce an external label to monitor the variable learning when conditioning the response generation on a specific attribute(s). In addition, we also propose to keep two separate dialogue contexts for each speaker in the conversation, in order to learn the speaker-aware information like personality, sentiment, styles, etc. The experimental results demonstrate that our framework is able to generate responses conditioned on specific attributes which is contributing to utterance-level coherence.

The second part (work 3 and 4) explores solutions for problem 2 with the aim of improving the conversation-level coherence. Note that conversation is unique in that information will change as the conversation goes, and the information at the current state depends on both the current utterance and previous information states. Therefore in the works in this part, we put efforts to explore the dynamics of information to improve conversation-level coherence for social chatbots. Specifically, in work 3, we leverage the meta-path information and propose a meta-path-augmented chatbot which firstly compares the context vector with each of the learned meta-path vectors, and then selects the candidate entity(s) that complies with the most similar meta-path. In work 4, we identify social coherence and individual coherence as two intention factors in conversation modeling, and design two strategies to in-

corporate them for multi-round response generation. On two real-world multi-round conversation datasets, we demonstrate the effectiveness of the proposed approach in improving conversation-level coherence.

The third part (work 5 and 6) delves into problem 3 and investigates how to improve the context-level coherence. Despite the recent improvements, the majority of existing methods learn the conversation representation and the information representation separately, which creates an obstacle for the chatbots to accurately model the conversation context and in turn influences the response quality. In the last part of our work, we argue to regard both conversation utterances and other information as a whole conversation context, and propose structured models to integrate the potential interactions among the conversation context. In work 5, we unify conversation utterances and background knowledge in one graph , and establish an innovative graph encoder to learn finer and deeper features for better response generation. In work 6, we together consider the emotion and intention states of the speakers, and propose an adversarial-augmented hierarchical model to generate responses that are sensitive to speaker states. Through extensive experiments, we verify the hypothesis that human emotions put a prior effect on conversation behavior.

In summary, we study the problem of open-domain conversation modeling and response coherence in a systematic way. We demonstrate the effectiveness of the proposed approaches on real-world datasets, which implies the potentials of our works when applying in real-world scenarios, such as empathetic companions for the elderly and entertaining social chatbot.

Publications Arising from the Thesis

Journal Papers

1. **Yanran Li**, Ruixiang Zhang, Wenjie Li, and Ziqiang Cao. Hierarchical Prediction and Adversarial Learning for Conditional Response Generation. IEEE Transactions on Knowledge and Data Engineering. 2020. (TKDE)

2. **Yanran Li**, Wenjie Li, Ziqiang Cao, and Chengyao Chen. incorporating relevant knowledge in context modeling and response generation Computational Linguistics. 2019. (CL, Under Review)

Conference Papers

3. **Yanran Li**, and Wenjie Li. Meta-path Augmented Response Generation. 2019. (AAAI' 2019)

4. **Yanran Li**, Hui Su, Xiaoyu Shen, and Wenjie Li. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. 2017. (IJCNLP' 2017)

5. Tong Che*, **Yanran Li***, Athul Jacob, Yoshua Bengio, and Wenjie Li. Mode Regularized Generative Adversarial Networks. 2017. (ICLR' 2017, * Equal Contribution)

6. **Yanran Li**, Wenjie Li, and Zhitao Wang. Improving Knowledge-aware Response Generation with Graph-Structured Context Understanding. 2021.

(SIGIR'2021)

7. **Yanran Li**, Chengyao Chen, and Wenjie Li. Modeling Social and Individual Coherence For Multi-round Response Generation. 2021. (EACL'2021, Under Review)

8. Jiayi Zhang, Zhi Cui, **Yanran Li**, Chen Wei, Jianwei Cui, and Bin Wang. Writing Polishment with Simile: Task, Dataset and A Neural Approach. 2021. (AAAI' 2021)

9. Zhi Cui, **Yanran Li**, Jiayi Zhang, Jianwei Cui, Chen Wei, and Bin Wang. Focus Constrained Attention Mechanism for CVAE-based Response Generation. 2020. (EMNLP' 2020 Findings)

10. Shuo Wang, **Yanran Li**, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. $PM_{2.5}$-GNN: A Domain Knowledge Enhanced Graph Neural Network For $PM_{2.5}$ Forecasting. 2020. (SIGSPATIAL' 2020)

11. Jianbo Ye, **Yanran Li**, Zhaohui Wu, James Z. Wang, Jia Li, and Wenjie Li. Determining Gains Acquired from Word Embedding Quantitatively using Discrete Distribution Clustering. 2017. (ACL' 2017)

12. Xiaoyu Shen, Hui Su, **Yanran Li**, Wenjie Li, Shuzi Niu, and Akiko Aizawa. A Conditional Variational Framework for Dialog Generation. 2017. (ACL' 2017)

Acknowledgements

First and foremost I want to thank my supervisor, Prof. Li Wenjie, Maggie. Academically, I received helpful critiques and patient guidance on conducting research from her. Personally, I learned essential qualities of an optimistic and dedicated person from her. Whenever I get lost, she is always there for help. Without her continuous support and enthusiastic encouragement, I would not survive my PhD journey. I feel lucky and honored to be her student.

I also appreciate all our lab members, Chen Chengyao, Zhitao Wang, Cao Ziqiang, Lei Yu, Chen Qiang and etc, for their thought-provoking discussions on my research. I also want to express my appreciation to my office mates, Dan Xiong, Rong Xiang, and Li Minglei, who gave me kind help in my Ph.D. life. My appreciations are also given to my dearest friend Saboya Yang for her continuous concern and encouragement.

Last but not the least, I dedicate this dissertation to my family for all their constant and unconditional love. My deepest gratitude is always given to my parents, who raised me with all their care and supported me to pursue my own dreams. And most of all, I have no words for how incredibly I learn from my loving and supportive family members. They gave me rebirth at the end of this hard journey, and showed me the faith at the new beginning of my life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

With the popularity of social networks and mobile applications, people share information in more different ways, which are often in the form of dialogues. Since dialogues are more natural manners to communicate and exchange information, both researchers and industrial developers have paid high attention on investigating dialogues, especially on building intelligent conversational agents. The primary goal is to better understand the semantics during conversations and provide natural and effective service through turns of communications.

Owing to the development of deep learning technologies, the pioneering work on developing intelligent dialogue systems often relies on a neural encoder-decoder architecture [218, 208, 190, 252]. According to the application scenarios, these dialogue systems can be roughly categorized into two classes: task-oriented dialogue systems and non-task-oriented social chatbots. The former class targets at assisting human users to effectively accomplish a set of pre-defined tasks, e.g., information request, restaurant booking, flight checking, hotel accommodation, and etc. Differently, non-task-oriented conversation systems, a.k.a. social chatbots, are to accompany humans and build amicable relations with humans by freely conversing with people on open-domain topics. In this thesis, we focus on building the latter type, open-domain

social chatbots.

The typical task of building open-domain chatbots is response generation, the task that requires chatbots to generate high-quality responses given user inputs. In recent years, neural response generation has achieved significant success in both academic [190, 313, 304, 185] and commercial worlds [200, 175, 315]. However, many challenges still remain. Most typically, the generated responses are often less informative and coherent to the conversation context. For example, these chatbots tend to generate dull and meaningless responses like "I can't tell you", "I'm not sure", "I think so" [101]. Such responses are far from satisfactory.

From the perspective of human-computer interaction, some researchers have investigated the perceptions and expectations regarding the use of conversational agents, and pointed out the importance of response naturalness, i.e., human-like qualities [21]. The most frequently mentioned components comprising conversational agent's naturalness are: responding informativeness [129, 140] and coherence with the preceding context [85, 154]. Obviously, generating meaningless responses like "I'm not sure" is under the user expectations for open-domain chatbots.

In the field of neural conversational modeling, researchers term the problem as *generic response problem* or *safe response problem*. The causes of this problem are multi-folds. The most fundamental one points to the underlying architecture adopted by these chatbots. Typically, the majority of text generation approaches in these studies are borrowed from neural translation [8], which applied the sequence-to-sequence(seq2seq) architecture [218] on a large scale of parallel corpora. Based on the architecture, neural generative models are trained to learn the post-response mappings using maximum likelihood (MLE) training objective. This kind of objective induces the model to treat the post-response relationship as one-to-one mappings. However, the conversations in the real world often embodies one-to-many relationships, where a post is often associated with multiple valid responses [310]. Such

discrepancy is one of the fatal causes to the generic response issue.

To tackle this issue, there are several lines of mainstream approaches. Previous studies modify objective functions to introduce diversity-promoting factors [104, 106, 123]. The following work encourages diversity of responses during beam search [101, 231], or introduces random factors by sampling [74, 2] rather than generating highest likelihood sentences.

This thesis falls into another popular research line that incorporates extra information like topic, cue-word, or style to encourage informative responses [150, 269, 239, 287, 52].Incorporating extra information is motivated based on the following observations. Firstly, conversations often heavily rely on background knowledge. For example, it is common for humans to mention some background articles they have read about the conversation topic. Secondly, dialogue is a dynamic information exchange flow[187]. It often consists of multi-turns of verbose utterances, where each utterance is dependent on what has been previously mentioned. This makes dialogues different from formal texts like news, and it is thus more challenging to understand conversations than plain texts. Thirdly, conversation is a social behavior formed by at least two parties. People produce utterances to reply and echo others, and when utterances are often turned from different interlocutors, it will lead to the topic drifts. More importantly, conversations are emphatically distinct in the ways that human consider others feelings and react properly. Considering these distinctions, it is a necessity to consider extra information when developing social chatbots to encourage the informativeness and coherence of the generated responses.

1.2 Research Problems

The primary goal of this thesis is to investigate the benefit of extra information for social chatbots, and how to effectively utilize different types of information to improve

response quality. The investigations are expected to handle the real-world problems shed lights on the development of social chatbots. The main research problems to be addressed in the thesis are listed as follows:

- *Problem 1: How to explore the benefits of extra information for conversation context modeling and response generation when building open-domain chatbots?*
- *Problem 2: How to develop an effective chatbot to learn the information change through turns of conversations and consider the dependencies among them?*
- *Problem 3: How to understand conversation context better through capturing the interactions between extra information and conversation utterances and improve conversation coherence in a holistic view?*

The first problem focuses only on exploring extra information for conversation modeling and requires a proposal of a minimum viable chatbot to examine their significance at the utterance level. Extra information can be viewed as an additional source of conversation context, which plays vital role for both conversation understanding and response generation. The main challenge of this problem is how to obtain additional information and design applicable models to make full use of the information in different types. A chatbot model equipped with more information is expected to better understand the conversation and respond more coherently.

The second problem pays attention to exploiting extra information in the multi-turn conversation setting. Typical types of information include background knowledge and speakers' internal states. As the conversation goes, these types of information will change upon the previous states, and in turn influences the next response. Hence, the reveal of the information dynamics is beneficial to improve the information prediction accuracy as well as response coherence. The main challenge of this problem is how to design effective mechanism that captures the information flow and

generate responses coherent to the information states.

The last problem emphasizes on the interaction between extra information and conversation utterances and requires for a unified model to comprehend them jointly. Different from the first and second problems, which only need to model a single source information for response generation, this problem aims at exploring the mutual impacts among the conversation context, which consists of multiple sources of information. Intuitively, when responding to others, humans will simultaneously consider various kinds of information to make the response coherent to the whole context. For this problem, the key challenge is how to together formulate multiple sources with different structures and how to include the interactions into the unified framework for holistic conversation modeling.

1.3 Research Overview and Contributions

Nowadays, Seq2Seq models have proven their great potentials in building conversational agents. Because dialogue context is essential for conversation modeling, researchers have proposed several methods for modeling dialogue context, especially history utterances, to improve the coherence of responses. However, conversation context is not limited to history utterances. It is also related to other information such as background knowledge and speaker information. To encourage response coherence, chatbots need to consider more types of conversation context as we humans do. Moreover, despite recent studies on response coherence, the progress of this trend is still at the initial stage of utterance-level coherence. There exist a lot of expectations and improvements to be achieved on building the intelligent conversational models.

In our work, we target at developing intelligent open-domain chatbots based on three levels of coherence, i.e., **utterance-level**, **conversation-level** and **context-**

level coherence. In accordance, the research works in this thesis are organized by three parts. The first part focuses on the utterance-level coherence through extra information injection, and the second part attempts to improve conversation-level coherence by introducing dependencies of extra information, while the third part concentrates on the context-level coherence by considering conversation and extra information in a holistic view. The overview of these works is summarized in Table 1.1. For utterance-level coherence, we establish two frameworks (work 1 and work 2) to solve the aforementioned research problem 1. For conversation-level, we propose two approaches (work 3 and work 4) to address the research problem 2. For context-level coherence, we develop two models (work 5 and work 6) to explore solutions for the research problem 3. The motivations and contributions of these works are briefly summarized in below.

Work 1 & 2: Utterance-level Coherence via Knowledge and Emotion Incorporation

Due to the insufficiency of vanilla Seq2Seq models in response generation, as initial attempts, we propose to incorporate two typical kinds of extra information into Seq2Seq models, i.e., background knowledge and speaker emotion. In work 1 and work 2, our research targets are to explore which type of knowledge/emotion is more suitable for existing chatbots to utilize, and how to make good use of the equipped knowledge/perceived emotion. To solve these problems, two novel chatbots, namely MIKE and SPHRED based on Seq2Seq models are proposed.

MIKE is a chatbot equipped with a knowledge base (KB) that learns to recognizes necessary knowledge relevant to the utterances, and generates coherent and entity-aware responses based on the detected knowledge. While previous works treat attributes and entities equally, our work is novel in discerning their differences and incorporating them in different manners. In this way, our chatbot MIKE better cap-

Table 1.1: Overview of Research Works in This Thesis

Research Work	Information Category	Research Problem	Publication Venue
Work 1: Incorporating Relevant Knowledge in Conversation Modeling	Knowledge	Problem 1	CL [111] (under review)
Work 2: A Conditional Variational Framework for Dialog Generation	Emotion	Problem 1	ACL [195] & IJCNLP [113]
Work 3: Meta-path Augmented Response Generation	Knowledge	Problem 2	AAAI [110]
Work 4: Social and Individual Coherence for Multi-round Response Generation	Intention & Knowledge	Problem 2	CIKM (under review)
Work 5: Graph-Structured Context for Knowledge-grounded Response Generation	Knowledge	Problem 3	SIGIR [112]
Work 6: Adversarial-augmented Hierarchical Prediction for Empathetic Response Generation	Emotion & Intention	Problem 3	TKDE [109]

tures the conversation logic with the help of contextual attributes, which in turn leads the respones to be more coherent. Regarding to emotion incorporation, the proposed SPHRED is a conditional conversational model allowing the responses to be controlled by specific attributes. Additionally, the two speakers in the conversation are associated with two separate encoders to model their dialog states while maintaining speaker-aware features. This model is flexible and potential to be applied to many scenarios for generating informative responses that are consistent with the specified attributes.

Contributions: In work 1, we propose to utilize contextual attributes and entities in their own ways. The contextual attributes contribute to capture the conversation logic for context modeling. The related entities are beneficial to generate responses when referring is needed. We develop a novel movie knowledge-grounded chatbot, namely MIKE, which firstly locates contextual knowledge from MKB that we build in advance, and then generates entity-aware responses based on the attribute-aware context representation. On two movie conversation corpus, our MIKE significantly outperforms other four knowledge-grounded models. This work is under the review (minor revision) of the *Computational Linguistics* [111].

In work 2, we present a conditional variational model for controlled response generation. The main idea is to introduce an external label to monitor the variable learning when conditioning the response generation on a specific attribute(s). In addition, we also propose to keep two separate dialogue contexts for each speaker in the conversation, in order to learn the speaker-aware information like personality, sentiment, styles, etc. The experimental results demonstrate the flexibility and potentials of our model. This work has been accepted by the *55th annual meeting of the Association for Computational Linguistics (ACL)* as a conference paper [195]. Also the developed dataset has been accepted by the *8th International Joint Conference on Natural Language Processing* [113] as a conference paper, and included as

a benchmark dataset by huggingface NLP library.[1]

Work 3 & 4: Conversation-level Coherence via Knowledge and Intention Incorporation

In work 1 & 2, we only incorporate extra information into utterances. Although history conversations are explored, the models in work 1 & 2 regard them as a single sequence of utterances without taking into consideration the dependency among the conversations. However, conversation is unique in that information will change as the conversation goes. The information at the current state depends on both the current utterance and previous information states. Therefore in our later works, we put efforts to explore the dynamics of information to improve conversation-level coherence for social chatbots.

Specifically, in work 3, we propose a meta-path-augmented chatbot namely MOCHA to capture the knowledge structure among the conversations. We assume that meta-paths over the mentioned entities are indicative of conversation flow, and it is thus reasonable to generate responses by leveraging the meta-path information. In particular, 10 most high-frequent meta-paths are defined according to the conversation data, and are then encoded into vectors for model use. Afterwards, MOCHA firstly compares the context vector with each of the learned meta-path vectors, and then selects the candidate entity(s) that complies with the most similar meta-path. In work 4, we develop CHEER to model communication intentions in social chatbots through capturing social coherence and individual coherence explicitly. Social coherence regards what the other speaker has said, and is captured by inter-speaker interactions between two adjacent utterances using a novel interaction unit. Individual coherence considers what the chatbot itself has been proposed, and is handled by keeping separate entity memories to ensure local consistency.

[1] `https://github.com/huggingface/nlp/pull/556`

Contributions:

In work 3, we propose to model conversation-level coherence by taking into account conversation flow for knowledge-grounded chatbots. We leverage meta-path information of entity mentions and propose chatbot MOCHA, which is augmented using meta-path information to have awareness of conversation flow. On two movie conversation corpus, our MOCHA significantly outperforms the compared models. To the best of our knowledge, our work is the first to explore meta-path information in social chatbots. This work has been accepted by the *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)* as a conference paper [110].

In work 4, we identify social coherence and individual coherence as two intention factors in conversation modeling, which have been largely neglected before. We propose a chatbot CHEER where two carefully designed strategies are introduced to incorporate these two kinds of coherence for multi-round response generation. On two real-world multi-round conversation datasets, we validate the effectiveness of the proposed approach and demonstrate the necessity of intention factors in coherence modeling. This work is under the review as a conference paper of the *16th conference of the European Chapter of the Association for Computational Linguistics (EACL).*

Work 5 & 6: Context-level Coherence via the Interaction between Information and Conversation

Although recent studies proposed several knowledge-grounded and emotion-aware conversational models and considered information structure in certain ways, they learn the conversation representation and the information representation separately. For example, the utterance representation is modeled by condensing the history information, and meanwhile the entity vector is learned to capture KB networks using graph embedding approaches. The blindness between utterance and entity vectors hinders the chatbots to precisely comprehend the conversation context and

thus harms the quality of the response. In the last part of our work, we argue to regard both conversation utterances and other information as a whole conversation context, and propose structured models to integrate the potential interactions among the conversation context.

In work 5, we unify conversation utterances and background knowledge and establish an innovative Context Graph Encoder (CGE) in order to represent the graph-structured knowledge-enhanced context in an integrated manner, which serves as basis for knowledge reasoning when generating responses. In work 6, we together consider the intention and emotion states of the speaker using two *discrete* variables, and employ a *continuous* variable to allow content-level diversity. To control models' behavior in more fine-grained way, we devise a adversarial learning objective and apply it on the variable-level. The ablation studies verify the novelty and effectiveness of the proposed adversarial-augmented hierarchical response generation.

Contributions: In work 5, we define and model the graph-structured conversation context derived from both history conversations and external knowledge. We develop a novel graph-based encoder, namely CGE, that enables holistic conversation understanding. Through extensive experiments, we demonstrate that the effectiveness of our approach as well as the contribution of the proposed graph encoder. This work is under the review as a conference paper of the *16th conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

In work 6, we consider the speaker's emotion and intention in one, single model by introducing two *discrete* variables to model them and employing one content-level variable to encourage the response diversity. The hierarchy among the three variables is explored and validated by comparing different model variants. In order to better improve the generation performance, the hierarchical model is further augmented with a variable-level adversarial learning objective. Based on the extensive experimental results, we find that human emotions put a prior effect on conversation be-

havior, especially on conversation intentions. We also emphasize the contribution of the devised adversarial learning which is devised and performed on the variable-level. The work has been accepted by IEEE Transactions on Transactions on Knowledge and Data Engineering (TKDE) as a journal paper [109].

1.4 Structure of Thesis

The thesis is organized as follows to give an overall picture. Chapter 1 firstly introduces the background of the researches on dialogue systems especially on open-domain neural response generation. Then this chapter also explains the three key problems, research overview and contribution of this thesis. Chapter 2 provides a comprehensive survey on the related work, including mainstream retrieval-based and generation-based approaches, context-aware models like knowledge-grounded chatbots. According to the three research problems, the thesis is divided into three parts. The first part (Chapter 3 and 4) mainly introduces basic chatbots that is able to utilize extra information. Chapter 3 presents a novel knowledge-aware chatbot, which utilizes both attributes and entities in structured knowledge bases. Chapter 4 investigates the effect of emotion for social chatbots and builds up a simple model that learns users' emotion and responses with specific emotions. The works presented in the second part (Chapter 5 and 6) are established based on the previous parts. Chapter 5 investigates how to incorporate additional knowledge information meta-path to capture the structure of knowledge utilization during the conversations. Without loss of generality, in Chapter 6, we explore the impact of the intention information on social chatbots and propose novel designs to model social and individual coherence when intention is implicit. Based on the findings in previous work, the third part (Chapter 7 and Chapter 8) considers the interaction between extra information and conversation utterances. Chapter 7 structures knowledge base and conversation

utterances into a holistic context graph and examines its significance on conversation modeling. Chapter 8 combines emotion and intention together and proposes a hierarchical model to capture their dependencies. Finally, the last chapter, Chapter 9 summarizes the proposed approaches, our findings, contributions as well as suggestions on future work.

Chapter 2

Literature Review

In this chapter, we go through the studies that are relevant to the research works in this thesis. We mainly focus on summarizing conversation modeling from the perspectives of tasks, evaluation as well as existing approaches. We also present the literature review on dialogue context modeling with the aim of improving response quality.

2.1 Conversation Tasks and Evaluation

The research on building automatic conversational agents has a long-standing history. Formally in existing human-computer systems, the user inputs an utterance as the query, and the system returns a response as the output. According to their application scenarios, conversational agents can be categorized into *task-driven dialogue systems* and *chit-chat conversational agents*. The former task-driven systems are designed for assisting people to complete particular tasks, ranging from flight scheduling to restaurant reservation. These tasks are basically established in various vertical domains and the conversational systems are tailored to fulfill user needs within these domains. The latter chit-chat conversational agents, in contrast, are non-task-oriented. Since they are usually designed for social chit-chats, this type of conversational agents are often abbreviated as chatbots. Different from task-oriented

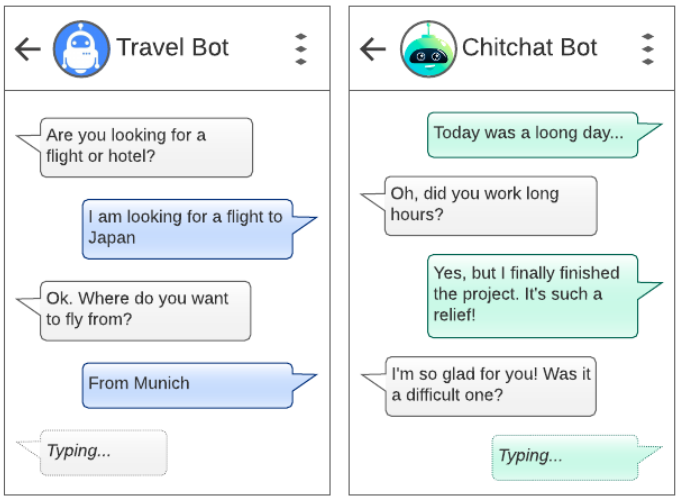

Figure 2.1: Examples of Two Conversation Tasks.

systems, chatbots aim to engage users in the open domain for entertainments, which makes it easier to go viral among users. With the development of digital economics, both task-oriented and open-domain conversational agents have garnered a lot of research attentions since they have greatly influenced human's daily life and demonstrated their huge commercial potentials.

2.1.1 Task-driven Dialogue Systems

The branch of task-driven dialogue systems is also called goal-oriented dialogue systems. As revealed by its name, task-driven dialogue systems are established to finish domain-specific tasks like travel request, restaurant booking, and so on. Before achieving the final goal, the systems need to learn the user need by communicating with users turn by turn. As such, to complete a task often involves a series of interactions between system and user. At last, these task-driven dialogue systems require a direct user feedback on whether the task is complete or not [59, 244, 179, 275, 286, 143].

In recent years, building task-driven dialogue systems have attracted enormous attention from enterprises because automatic systems are able to provide 24-hour ser-

vice and potentially bring in large commercial profits. For example, enterprises have launched intelligent personal assistants (IPAs) including Apple's Siri,[1] Microsoft's Cortana,[2] Google Assistant,[3] Facebook M,[4], Amazon's Alexa,[5] etc.

Traditionally, these dialog systems have been developed using heuristic rules and template-based approaches [247, 35, 196, 236]. Because it relies tedious effort to design rules and templates, these approaches constrain traditional task-oriented dialog agents to be applied within only small and specific domains, e.g., information query on transports [183]. Beyond rule- and template-based methods, modern task-oriented dialogue systems often consist of four components: natural language understanding (NLU) [244, 179, 275], dialogue state tracking (DST) [152, 99, 72, 143], dialogue policy learning (DPL) [253, 107, 211], and natural language generation (NLG) [59, 157, 235, 250]. With the development of deep neural networks, end-to-end approaches demonstrate their effectiveness on learning a fully data-driven task-specific dialogue systems [212, 255, 16, 252, 251].

Though achieving promising performances, end-to-end models usually depend on a considerable amount of labeled data, which prohibits them from easily applying to new and extended domains. Hence, it is worthy of exploring to transfer knowledge from a source domain with sufficient labeled data to a target domain with little labeled data. Existing work on multi-domain goal-oriented dialogue systems can be classified into two general categories. One category of work trains one, single model on the mixed multi-domain dataset [135, 47, 134]. Such methods make it to implicitly extract the shared features but fail to effectively capture domain-specific knowledge. The other category of work sets one separate model for each domain [249, 173].

[1] https://www.apple.com/ios/siri/

[2] https://www.microsoft.com/en-us/cortana/

[3] https://assistant.google.com/

[4] https://developers.facebook.com/blog/post/2016/04/12/

[5] https://developer.amazon.com/alexa/

Although supposed to better capture domain-specific features, this strand of works might ignore shared knowledge between different domains, and lead to performance degradation.

With the recent progress of the sequence-to-sequence (Seq2Seq) models in text generation [218, 8, 131], it becomes common to formulate task-oriented dialogue generation as the Seq2Seq mapping from the dialogue history to response [135, 47, 134]. This kind of modeling scheme makes the development of these systems no longer burdened with the manual design and laborious annotation of the modules. Notably, one of the remaining challenges is how to query the structured KB. Instead of calling APIs to interact with the KB, later approaches shift to integrate KB query as an input or processing module in the model. The most popular way is to regard the KB query as an attention network over the entire KB entities [47, 41, 134, 249, 180, 257]. Briefly speaking, the recent tread in building task-oriented dialogue systems follows the paradigm of modeling KB with dialogue systems in a unified manner, which is inspired from work on encoding extra information for Seq2Seq-based open-domain chatbots. We will detail the latter ones in the following sections.

2.1.2 Open-domain Chatbots

The other category of automatic dialog systems is chit-chat conversational agents, which aims at engaging and accompanying users by daily chatting. Such primary goal requires the chatbots to have a large open-domain conversation scope. Owing to the rise of deep learning techniques and the large amount of available conversation data [208, 193, 264, 305], we are now witnessing fast and encouraging progresses of chatbots in both industrial and academic fields [29, 277, 178, 51, 221].

In the research field of building open-domain chatbots, the mainstream approaches rely on the framework of neural Seq2Seq models [218, 209, 190]. However, these models alone are still inadequate to develop a satisfactory chatbot without extra control.

For example, standard Seq2Seq models tend to utter bland responses with little meaningful information. To tackle this issue, researchers explore to inject various information into Seq2Seq models. Among different kinds of information, the widely adopted ones are topic, personality, and emotion. Additionally, incorporating external background knowledge is another promising way to bridge the gap between an artificial dialogue system and a real human. We will detail these improvements in the next section.

In spite of these advances, Seq2Seq based chatbots still suffer from a lot of problems including understanding difficulty and response inconsistency. To address, Amazon Alexa Prize [178] provides a platform to collect real human-machine conversation data and pushes the research on social chatbots a step further [291]. The chatbots submitted to anticipate Amazon Alexa Prize are not necessarily using end-to-end solution. Rather, they are encouraged to focus on improving user experience and becoming a human-like companion. Ideally, a human expert conversationalist will blend a number of skills in a seamless way: During the conversation, he/she will provide engaging talking points, and listen to the other speakers. The human expert will also display knowledge, empathy and maintain a consistent persona [82, 201]. Likewise, to become a human virtual companion, it is necessary for social chatbots to acquire sufficiently high IQ, and to perform a range of skills in order to bond with and accompany the users in their daily life [200]. More importantly, social chatbots also need a sufficient EQ to cater for users' mental needs, such as emotional affection and social belonging, which are among the fundamental needs for human beings [137]. Therefore, large improvements can be made on building better intelligent chatbots that emphasizes desirable conversational skills. To step further on blending skills for social chatbots, [201] introduces Blended Skill Talk (BST), which pays attention to the desirable aspects by providing conversational context, i.e., topics and personas. Based on BST, [185] fine-tunes the models that make the conversational

model concentrating on desired aspects and skills, achieving large gains. Through extensive experiments, they find that small models trained using BST can match or even outperform larger models without BST.

The typical commercial social chatbot is Microsoft's XiaoIce [200].[6], and its rapidly increasing user indicates the necessity of chatbot service. The popularity of XiaoIce can be attributed to its unique design goal. As explained in [315], the primary goal of XiaoIce is to be an AI companion and form long-term, emotional connections with users. Such goal is different from early social chatbots since the latter ones do not care about long-term relationships. As emphasized in [315], the heart of XiaoIce's system design is the integration of both IQ and EQ. To implement this idea, XiaoIce is developed on an empathetic computing framework [25, 49] that enables it to feel human emotions and thoughts, and respond to users dynamically. Upon the empathetic computing framework, there is the core module of XiaoIce, i.e., Core Chat, to provide the fundamental communication capability. In specific, core Chat consists of two parts: (1) General Chat that is responsible for engaging in open-domain conversations that cover a wide range of topics; (2) Domain Chats that are responsible for engaging in deep conversations on specific domains such as music, movie and celebrity. Both General chat and domain chats are data-driven response generation systems which output responses in two stages: response candidate generation and ranking. As a hybrid commercial chatbot, the response candidates in XiaoIce are either retrieved from the databases which consist of human-generated conversations or texts, or generated on the fly by a neural response generator. By equipping the empathetic computing and Core chat modules, XiaoIce is able to take the text input and generate interpersonal responses.

[6] `https://www.msXiaoIce.com/`

2.1.3 Datasets and Evaluation

In early days, people establish conversational systems using hard-coded rules or human-written templates. Despite the simplicity, such methods require tedious human efforts on creating abundant rules or feasible templates to foster the agents. To remedy this issue, researchers explore data-driven methods and curate several datasets. Before this decade, the widely-known datasets in this research field include TRAINS [183], DBOX [168], bAbI synthetic dialog [18] and Movie Dialog datasets [43]. For example, bAbI [18] is a synthetic dialog dataset that consists of simple reasoning on objects. However, these datasets are of small size and the systems trained on these datasets are often limited within a certain domain.

Obviously, a natural conversation often goes out of scope easily which is problematic to domain-specific systems. For open-domain chatbots, it is critical to train the models on large-scale datasets. [191] proposes a neural responding machine (NRM) and evaluates its effectiveness using the newly introduced dataset Sina Weibo [240]. The post-response pairs in the dataset are constructed by collecting the posts and replies in the same threads on Weibo, the largest social network in China.[7] Similar social network-based datasets include Ubuntu [126], Twitter [184], Reddit [3], etc. There are also several Chinese datasets proposed in recent years, such as Douban Conversation Corpus [264] and E-commerce Dialogue Corpus [305]. In the latest year, [245] crawls 79M conversations from Weibo, and constructs Weibo dataset (LCCC-base) through a rigorous cleaning process. LCCC-base is then mixed with several public Chinese conversation datasets to obtain a larger Chinese conversation dataset (LCCC-large).

Despite the large scales of these datasets, the "conversations" from social networks are often noisy and short. Even worse, these "conversations" are inherently differ-

[7] `https://weibo.com/`

ent from real-world human conversations since they are synthesized by linking the online posts and replies. To facilitate the development of automatic conversational models, more and more dialogue datasets are curated using crowdsourcing to collect conversations between crowd workers. The dataset PERSONA-CHAT [102] assigns a persona for each crow worker and considers consistent personality while continuing a natural conversation. Modified from PERSONA-CHAT, Dialogue NLI [248] is a natural language inference dataset, which has been demonstrated useful to improve the conversation consistency. CoQA [181] is curated under the scheme of Wizard-of-Woz where two annotators are paired to converse given a passage by asking and answering questions towards the passage. Note that each question in the passage is contextually related to the history utterances. More recently, there are a handful of knowledge-grounded corpora proposed using different sources under different languages. Unlike open-domain chit-chats, researchers develop knowledge-grounded corpora by specifying some topics which need extra knowledge during conversation. Some datasets [313, 54, 122, 225, 172] collect dialogues and label the knowledge annotations using string matching, named entity recognition, and other linguistic-driven techniques. In particular, CMU DoG [313] utilizes 30 Wikipedia articles about popular movies as grounded documents. Another dataset also using Wikipedia is Wizard of Wikipedia (WoW) [42], which covers up to 1,365 dialogue topics and is much larger than CMU DoG. Considering its topic coverage, WoW emphasizes more on the generalization ability of conversation models. In addition to the unstructured text, India DoG [146] uses fact tables as background resources, OpenDialKG [147] and DuConv [260] build up their corpora based on structured knowledge graphs.

Evaluation is another long-standing issue in developing social chatbots. Generally, automatic measurements like N-gram matching are widely-adopted criteria when evaluating the response quality. Existing metrics like BLEU [161] and ROUGE [114] usually calculate the overlapping of the words between the reference and the candi-

date on the surface level. In addition, [101] proposes Dist-1 and Dist-2 scores for the ratios of the uni-grams and bi-grams, to indicate the informativeness of the responses. Notably, these metrics only consider word-level similarity and often fail to capture the real semantics under the compared text. However, chatbots are open-ended and do not necessarily solve a clearly-defined task. It is unsafe to assess the chatbot performances simply based on a pre-defined ground-truth. As a result, these N-gram based metrics inevitably show poor correlation with human judgment [119, 156, 26]. To better evaluate text generation models, different neural network-based metrics are proposed. BERTSCORE [301] develops a soft approximation of context embedding to replace the hard N-gram matching. MOVERSCORE [308] applies word embeddings of a pre-trained model to find the semantic similarity via Word Mover's Distance. Most recently, [65] proposes PERCEPTION SCORE, a system-level automated evaluation metric that learns the difference between the generations and the distribution of references. Despite the improvements, these methods often utilize prior knowledge in a large pre-trained neural model, which neglects the fact that semantic meaning of generation and reference are context-dependent, especially in open-ended generation task. Considering the limits of existing studies, researchers often evaluate the conversational models by calibrating human judgments and automatic evaluation metrics in order to truly fathom the quality of generations.

2.2 Existing Approaches

We will introduce the mainstream approaches for building open-domain chatbots in this section. According to implementation methods, conversational systems can be typically classified into *retrieval-based models* and *generation-based models*. Based on the massive available data, it is straightforward to build a dialogue system using information retrieval techniques. In this case, the problem is formulated as given a

user query, the system searches for candidate responses by matching metrics. TAnother way to adopt language generation approaches and build a generation-based conversational system alternatively. Inteading of retrieving an existing reply candidate, these approaches generate a response using a language model trained before. With deep learning techniques applied, both retrieval-based generation-based systems are greatly advanced and generally are developed upon the sequence-to-sequence architecture. Additionally, ensembling retrieval-based and generation-based approaches together has also emerged as a powerful choice to build conversational agents. Whereas retrieval-based approaches are able to provide human-written high-quality response candidates, generation-based approaches are flexible to refine the response candidates into more suitable ones. By combining the benefits from the two worlds, ensembling methods are also appealing and their responses are often more preferred in terms of fluency and relevance.

2.2.1 Retrieval-based Approaches

Retrieval-based approaches select proper responses from data pool by matching techniques [240, 87, 126, 278, 318, 268, 264]. Retrieval-based chatbots are often composed of two components, retrieve module and re-rank module. With regarding to retrieve module, encoding queries and responses into vectors with same semantic space are first constructed, and then nearest neighbor search (NNS) within vector space is the following step. Many approximative algorithms [193, 198] have been examined to improve retrieve efficiency. In the research field, focuses have been put more on the re-rank module [160, 237, 219, 318] that selects the most appropriate response from some candidates.

Typically, matching models are adopted for the rerank module in order to capture query-response semantic relevance. Early studies on matching model could be classified into two main categories: matching function learning and representation

learning. Approaches of matching function learning first represent queries and responses based on shallow features, and then apply some deep models to discover the matching patterns. The representative methods include ARC-II [75], MatchPyramid [160], Match-SRNN [237], RCNN [166]. One thing to note is that the query-response interaction at the beginning requires huge time cost, which is a drawback for online environments. On the other hand, methods of representation learning start with deep models to acquire representations for query and response, and in the last step, they often use simple matching function like dot-product and cosine distance to measure the semantic similarity [158, 238, 176].

Current state-of-the-art methods on response retrieval follow a representation-interaction-aggregation framework [263, 319]. For each utterance-response pair, matching signals are distilled from their interactions based on their representations, and then are aggregated as a matching score. This line of methods has two shortcomings. First, representations learned by these methods work well for the re-ranking task but fail in the semantic search task with large-scale responses, which impede their potentials in online environments. Second, although utterance-response interaction has proven to be crucial to the performance of the matching models [264], the interaction is often executed in a rather shallow manner where matching between an utterance and a response candidate is determined only by one step of interaction. To alleviate the latter shortcoming, [222] increases the depth of context-response interaction in matching and shows that depth can bring significant improvement to model performance on the task of matching-based response retrieval.

Owing to the fast development of pre-training techniques [40, 125], a machine is able to achieve promising performances which are sometimes very close to human performance. [73] makes the first attempt to apply pre-trained language models in response selection where multi-turn conversations exist. Based on the pre-training models, [62] proposes Speaker-Aware BERT where the model is designed to have

awareness of the speaker change, which is an instrumental nature inherited in multi-turn conversations.

2.2.2 Generation-based Approaches

To date, generation-based approaches have been demonstrated effectively in dialogue modeling and response generation. [233] pioneers this direction by originally applying the Seq2Seq model in open-domain response generation. [191] follows and proposes Neural Responding Machine (NRM) with several context generators. Despite the potentials, vanilla Seq2Seq models are observed to generate generic and dull responses, such as "*I don't know*" or "*I'm OK*" [101]. This is called the "generic response problem" or "safe response problem". Literature in the past few years has identified a number of reasons for this problem. Accordingly, several lines of research methods have been proposed to tackle this problem.

It is worth-noting that one important and maybe the most deeply-rooted cause lies in the nature of the generation architecture. Although the sequence-to-sequence (Seq2Seq) architecture [218] has been broadly utilized for response generation in Short-Text Conversation [233, 190], it was originally designed for Machine Translation to model one-to-one mapping for identical semantics expressed in two different languages. However, this one-to-one relationship goes against the nature of conversation, i.e., multiple valid responses exist for a given post, namely one-to-many relationship [311]. To tackle this issue, some intial works modify the decoding strategy to improve response quality. Researchers have proposed new objectives [101], enhanced decoding algorithms [104]. For example, [101] proposes to penalize dull and bland responses based on the maximum mutual information (MMI) during the beam searching. The following works adjust the data distributions by using different weighting methods to sample the data in order to force the model to emphasize more on the rare samples [153, 123]. The majority of these approaches make effort in either

the stage of pre-processing or the stage of post-processing in the testing phase. In other words, these methods do not pay attention to altering the architecture of the Seq2Seq models.

Later, many attentions have shifted to incorporate useful information into conversational models to improve the diversity and informativeness of the generated responses. One line of research introduces a set of latent responding mechanisms and generates responses based on a selected mechanism. [310] learns the post-response mappings as a mixture of the mechanisms, but it is questionable that they only rely on one single mechanism when generating responses given a new post. [28] adopts posterior selection to build one-to-one mapping relationship between the mechanisms and target responses. Another line of research adopts Conditional Variational Autoencoder (CVAE) to introduce latent variables into Seq2Seq models through variational learning. The latent variables are supposed to capture the discourse-level semantics of target response and in turn encourage the response informativeness. Recent literature along this line attempts to improve the model performance by putting extra control on the latent variable [306, 66, 52]. Despite the control, these methods still rely on the discourse-level latent variable, which is too coarse for the decoders to mine sufficient guiding signals at each generation step. As a result, these variational models are observed to ignore the latent variable [306, 66, 52] and to generate semantically irrelevant or grammatically disfluent responses [174]. In this thesis, we explore different manners of approaches to develop generalized conversation models that are capable of incorporating various information effectively.

There is another issue that affects the performance of generation-based models, which does not apply to retrieval-based systems. At the inference mode, generative models must select a decoding method to synthesize the response word by word. The choice of decoding algorithm is crucial, and models with different decoding algorithms may produce completely different results. In particular, different decoding algorithms

usually prefer different lengths of the generated responses, which is crucial for human judgments. For example, previous work has also reported that beam search is not as effective as sampling [74, 2]. However, as shown in a recent study, calibrating the hype-rparameters carefully can provide powerful results by weighing trade-offs.

In the past decade, the academic community has witnessed the flourishing of generation-based methods, especially when the model of pre-training language models has recently been proposed. For example, DialoGPT [304], as a typical pre-trained dialogue model, is trained over 147 million dialogue-like communication trainings on Reddit after 12 years. As a result, DialoGPT has achieved close to human performance in both automatic and manual evaluation in a single-round dialogue setting. Many of the following works in this line show that generative dialogue systems using pre-training techniques can produce more relevant, satisfying, and contextual responses than before. Meena [2] depends on the Evolved Transformer [202] and [108] studies the generation of dialogue by fine-tuning Chinese GPT on some small dialogue datasets. [245] proposes a pre-trained dialogue model CDial for Chinese dialogue generation, which is trained on a collection of multiple large Chinese dialogue datasets.

2.2.3 Other Approaches

Besides Seq2Seq based approaches, ensembling techniques, adversarial learning and reinforcement learning algorithms have also been attempted to improve the response quality.

Adversarial learning [61] is firstly introduced in dialogue systems for evaluation in [89]. By their study, they train a discriminator as a proxy to differentiate the generated responses from real responses in terms of length, genuine and diversity. After that, [106] and [276] aim to apply generative adversarial networks [61] to alleviate the "safe response problem" [101, 276]. These works adopt Seq2Seq models

as the underlying architecture, and add extra modules to enhance the generation procedure. To directly apply the GAN framework in response generation is non-trivial. [106] adopts reinforcement learning approach and approximately computed the rewards. [276] sidesteps the issue by using an approximate layer to replace the procedure of discrete sampling. [302] deploys a method based on adversarial learning and optimize the MMI objective directly in the model training phase [101]. These models adopt the Seq2Seq models as the generator and pay attention to the design of the discriminator, and at last the generators and discriminators in these works are jointly optimized. One of our research works [109] develops a hierarchical variational model augmented with a variable-level adversarial learning objective, which is able to produce outputs by hierarchically predicting the necessary speaker states, and then conveying the states in the final responses.

Deep reinforcement learning are widely used for policy learning in the goal-oriented dialogue systems[210, 118]. Recent research works also attempt to apply reinforcement learning to train neural dialogue models [105, 106]. However, previous approaches depend on the REINFORCE algorithm for model learning, which is known for being slow, unstable, and with high variance when rewards are sparse and delayed until the end of a task episode [144]. Later work [142] attempts to adopt the actor-critic method [7] to overcome these weaknesses. In terms of learning a dialogue policy for open-domain dialogue, [272] designs a policy network to predict dialogue acts and feed those acts into a response generation model to control responses. [70] designs a policy that integrates knowledge with dialogue acts at a sentence-level, and demonstrate that a basic rule-based dialogue policy can result in strong performance.

Recently, some researchers have ensembled several aforementioned methods to enhance the performance by combining each of their benefits. For example, it is feasible to rank two types of responses and return the top-1 result. Another reasonable and promising way to ensemble methods is to feed retrieved responses to a

generation-based model to enhance the informativeness and diversity of the generated response. [175] proposes that when the top retrieved response achieved a score above a certain threshold, it should be taken as the final response; otherwise, the response should be obtained from a generation-based model. Similarly, [205] reranks the two kinds of responses but firstly concatenates the retrieved responses into the context to generate a response. [262] designs a response-editing model that modifies a prototype using guidance from an edit vector. [320] and [299] cast response generation as a reinforcement learning process. Recently, a skeleton-then-response framework has been shown promising results for this task [23]. Nevertheless, how to precisely extract a skeleton and how to effectively train a retrieval-guided response generator are still challenging. [24] extracts the skeleton by an interpretable matching model and in their work, the skeleton-guided response generation is accomplished by a separately trained generator.

2.3 Context Modeling

A good conversational agent is expected to produce responses that are grammatically fluent, semantically relevant, and contextually coherent. However, as stated above, standard Seq2Seq models are prone to produce generic and bland responses [101]. Especially for multi-turn conversations where there are several turns of previous utterances, the responses generated by standard Seq2Seq models are often improper to the conversation context [110]. To mitigate performance gap and improve response coherence, how to effectively model conversation context becomes a more and more important issue.

At the beginning of applying neural networks for chatbots, researchers' efforts are mainly paid on exploring effective mechanisms to summarize conversation histories, i.e., the previous utterances before the current input. As the development of

neural chatbots, researchers have realized that conversation context is not limited to history utterances. Hence, later and recent attempts have been made to improve the capacities of conversational models by considering other context-related information such as external background knowledge and internal context features. In the following, we will review literature work on improving response coherence by grounding on external knowledge. One one hand, there exists a handful of literature attempting to enhance context understanding by leveraging additional contents [54, 42, 313]. On the other hand, a plethora of literature augments the decoder to leverage structured knowledge bases (KBs) [322, 122, 111, 289]. We will also review the studies considering contextual information when modeling conversation context, such as emotion [312, 117, 109], dialog act [307], speaker personality [102, 171, 300, 133, 27, 204], as well as topic [268, 150, 67]. The following contents will present a detailed review of modeling conversation contexts including dialogue histories, external knowledge as well as other related information.

2.3.1 History-aware Models

Generally speaking, context modeling has been a long-standing challenge in conversation research. For multi-turn conversations, typical conversation context includes history dialogues from previous conversation turns. Early efforts exploit history utterances as the only source of the conversation context, and draw on neural networks for their powerful representation capabilities [208, 189, 19, 188, 31]. Although [209] is conducted under retrieval-based approaches, it demonstrates clear improvement on the response quality when the context feature is integrated. After examining its benefit, the remaining problem is how to effectively utilize contextual information. [103] argue to model the words and utterances in different levels of hierarchy when learning the representations. Then, hierarchical representation learning is proposed to model the context, which discerns the word- and utterance-level information and

produce their representations in different ways. Both retrieval-based and generation-based conversational agents exploit and integrate the information from the two levels together.

In terms of retrieval-based systems, one line of approaches encodes multiple utterances of the context into a single context vector and uses the vector to match responses. [208] splices utterances into one sentence and then encodes it with LSTM. [318] takes each utterance as a unit and uses hierarchical GRU to encode utterances into a context vector to catch utterance-level discourse information. [63] considers an interaction between the context and the response in order to produce more descriptive representation, and uses an attentive hierarchical recurrent encoder to characterize the representation.

With regard to generation-based systems, [209] directly feeds embeddings of previous conversation turns with current inputs into the hidden layers of encoders. [208] develops a RNN based model HRED, which builds up a context encoder (ContextRNN) on the top of a word-level encoder (EncoderRNN) to hierarchically model the dialogue context. In order to include more variations and encourage diversity in generation, HRED is further extended in [189] with an extra continuous variable. Later, [19] uses Memory Networks to encode the unstructured history dialogues, while [68] utilizes structured knowledge in addition to the unstructured history. [188] expands the generation process by adding a sequence of discrete stochastic variables for each utterance, which helps generate responses with high-level compositional structure.

2.3.2 Knowledge-aware Models

Endowing the chatbots with extra background knowledge is another promising idea to improve response coherence. It is very likely that people respond in conversation focusing on certain topics rather on rambling among unrelated issues. To control the generation performance, researchers propose different methods to introduce various

contents into generation models. [150] is the first attempt to explicitly control text generation by selecting cue words based on the external Pointwise Mutual Information (PMI). Following their work, the improved approach has been moved to a more implicit and flexible manner [288]. Later work exploits other content types including topic, knowledge, etc.

Initial work on introducing topic into response generation models is to incorporate a topic variable calculated by Latent Dirichlet Allocation (LDA) [14]. Such topic information can either be learned from current input in conversation [268, 150], or obtained from conversation history as a prior knowledge [128]. [268] encodes both the input word embeddings and the topic keyword embeddings into a content encoder and a topic encoder, respectively. These two encoders then interact with each other in a joint attention mechanism to together determine the response decoding. However, it is hard to ensure that the topics learned from the external corpus are consistent with that in the conversation corpus. [128] comprises all previous dialog turns as a topic vector, which is then concatenated with hidden states to predict the response tokens to be generated. [67] recognizes both topics and keywords to evaluate conversation models using topic-centered metrics.

Nowadays, one of the most popular research directions is to build knowledge-grounded chatbots. [295] handles possible breakdowns in dialog systems by retrieving a short description to generate sentences. Some works attempt to incorporate implicit knowledge into chatbots to address the "generic response problem". [234] aligns topic-related descriptions from Wikipedia pages with Reddit comment threads, and proposes a coupling network to fuse the implicit knowledge before generating comments. While [150] and [268] inject conversation topics into Seq2Seq models, [67] recognizes topics and keywords to build up topic-based assessment scores. There also exist other content introducing models exploiting different types of knowledge. For example, [271] uses meta-words, and [321] utilizes the retrieved existing dialogues.

However, the leading cause of generating generic responses is that the model can not obtain enough background knowledge from the query message [54, 125]. To alleviate the lack of background knowledge, researchers have begun to introduce external knowledge into the generation. The knowledge can be unstructured knowledge texts [54], structured knowledge graphs [313], or hybrid of them [125]. Especially, [54] utilizes unstructured textual information as explicit knowledge for the chatbots. [254] presents a model allowing developers to express domain knowledge via software and action templates. [322] develops a dialogue system to talk about musics. Although grounding on structured knowledge, their system is focusing more on answering music-related questions.

The structured knowledge has the best quality, because it is generally extracted and summarized by the human. The structured knowledge graph can be either domain-specific [322, 122] or open-domain [290, 313]. Two previous studies [290, 313] have proved the feasibility of introducing commonsense knowledge into dialogue systems, where [290] is designed for retrieval-based systems, and CCM [313] is for response generation models. Following CCM, ConKADI [259] is designated to be aware of the context when using the knowledge, and it uses human's responses as posterior knowledge in training. Concurrently, [258] exhibits the central topic fact of a generated response, and it is controllable such that TopicKA can generate multiple diverse responses based on different topic facts. Especially, the majority of existing knowledge-grounded chatbot augments the decoder to leverage structured knowledge bases (KBs) [322, 122, 111, 289]. In addition to facilitating response generation, in this thesis, we also explore the benefits of leveraging KB for context understanding. [54] and [42] feed unstructured knowledge to the RNN-based and Transformer-based encoders. [313] and [122] combine factual embeddings with the encoder states. Different from the previous "shallow" combination approaches, in this thesis, we define and model the conversation context by combining history conversations and external

knowledge, by which we develop a novel graph-based encoder that enables holistic conversation understanding and in turn facilitates response coherence.

2.3.3 Other Context-aware Models

Learning the dialogue features explicitly is another way to model the coherence and diversity of the response. Among all the context-related features, identity attributes, speaker emotions and communication behaviors are widely explored. In the research line of modeling speaker identities, [102] introduces an extra variable representing personal information to capture personalized communication styles. [145] uses transfer learning techniques to train a personalized dialogue system. [3] also considers author information in Reddit Forum and modeled it as an input feature. More recent work considers the differences between the two speakers in a dialogue and models them separately [195]. [171] endows chatbots with a pre-defined agent profile to improve the coherence in the generated responses.

In addition to speaker's persona, there are also other internal factors affecting people's daily communication, among which emotion and intention are two essential ones. As a critical kind of intelligence in humans, emotional intelligence is potentially useful in building conversational agents. As studies have shown, endowing the conversational agents with emotional intelligence will make users engage longer and deeper, and thus improve the user satisfaction towards the chatbots [98]. When one find that his/her friend is upset, he/she usually will express his/her concerns toward the friend by asking the reasons behind the mood [139]. People often have shared mental states and these states together contribute greatly to form human's emotional intelligence, which consequently affect human's conversation behaviors.

To build emotion-aware chatbots, researchers' attention has been paid to detect emotion revealed in the history utterances and has been recently moved towards generating responses with the specific properties like sentiments, tenses, or emotions.

[79] proposes a text generation model based on variational autoencoders (VAEs) to produce sentences presenting a given sentiment or tense. [56] presents a RNN-based language model to generate emotional sentences conditioned on their affect categories. This study focuses on general text generations rather than in the case of conversations. [316] has collected a large number of Twitter conversations including emojis, on which they use emojis to express emotions in the generated text by trying various variants of the conditional VAEs. The representative work in this area is Emotional Chatting Machine (ECM) [312]. On the basis of the Seq2Seq framework, ECM and its subsequent methods [36, 207] mainly represent the given emotion category as a vector, and use a gating mechanism to add it to the decoding step to affect the response generation process. As experiments have shown, this approach will aggravate the issue of generic responses in some cases [109]

Another line of research work focuses on modeling communication behavior expressed as dialog acts in open-domain chatbots. [307] captures the diversity of responses through conditional variational autoencoder and utilizes dialog act information as the latent variable. There also exist certain literature focusing on generating responses conditioned on latent factors, which are less interpretable. [285] attempts to improve the specificity with the reinforcement learning framework by using the averaged IDF score of the words in the response as a reward. [311] introduces latent responding factors to the Seq2Seq model to avoid generating safe responses. However, these latent factors are hard to decide the number. In this thesis, we consider together the effect of emotion and intention on conversation modeling, and propose a hierarchical generation model to explore the dependency of these two factors in response generation.

Part I
Utterance-level Coherence

Chapter 3

Knowledge Incorporation for Utterance-level Coherence

3.1 Introduction

Different from task-oriented dialogue assistants, social chatbots are not necessarily to solve problems. Rather, they are designed to engage and company users by chit-chat conversations [200]. However, it is non-trivial to build a satisfactory human-like chatbots since open-domain chatbots are expected to have a large conversation scope. To constrain the research scope, knowledge-grounded conversation modeling becomes a new research direction. On one hand, unlike open domain dialogue, it involves some specific topics which need extra knowledge during response generation. On the other hand, it also has various indirect information to the topic such as chitchat, expressing a personal experience or opinions related to the topic, etc.

Therefore, developing knowledge-aware chatbots are beneficial for research on human-like conversations. To achieve this, it is instrumental for these chatbots to be equipped with conversation-related knowledge, a.k.a. contextual knowledge. When people are discussing on a movie, for example, they often express their attitudes towards the actors and the directors of the film. Previous literature on equipping chatbots with external knowledge often utilize unstructured knowledge. However,

there remain at least two issues unexplored: (1) Which type of knowledge is more suitable for existing chatbots to utilize? (2) How to make good use of the equipped knowledge? To explore these two problems, we equip the chatbot with structured knowledge base (KB). To better utilize KB for chatbots, we identify and exploit the following information in KB:

- The first is attributes in KB. Essentially, contextual knowledge is beneficial for conversation context modeling. People often start and keep a conversation following a certain logic. See a real example in Figure 3.1. Given the film *The Notebook* as the topic, user B imagines the new film *Spotlight* because user A is talking about series of romance movies acted by the actress Rachel. In the figure, bold words indicate the underlying attribute, which are illustrated as the red arrows linking the films. As the **attribute** of the film, actress holds as the underlying logic link that naturally guarantees the coherence when moving forward the conversation. It is thus reasonable to equip chatbots with the ability to recognize the underlying attribute(s) for conversation understanding and link to related knowledge based on the recognized attribute(s).

- The second is entities in KB. As another kind of contextual knowledge, **entities** are more important because they are extensively involved especially when people offer new information, provide supporting evidence, or refer to what has been mentioned. To facilitate response generation, chatbots need to also bear in mind related entities as candidates to be selected when responding to users. As revealed in the example, person A does not insist on the romance movies but moves on to the new one after person B introduces *Spotlight*. In regard of the current context, the entities being considered in each turn may change. The larger the number of the candidate entities, the harder it will be for the chatbots to reason the most suitable one based on the current context.

Person	Utterances
A	The **actress** is so amazing in starring also *The Notebook, About Time, The Vow*.
B	And the last year winner *Spotlight*. Love **Rachel** so much!
A	So **she** stars in *Spotlight*! I've yet to see it. This **actress** is gorgeous!

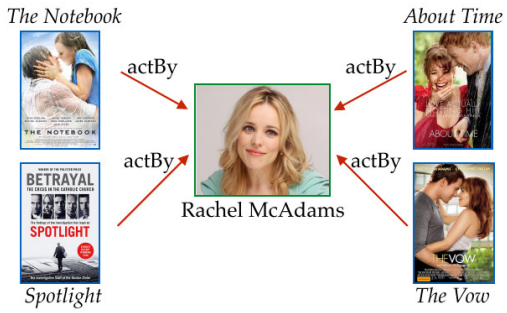

Figure 3.1: A Motivating Conversation Example.

To ease this issue, our idea is to selectively collect the candidate entities using the recognized attributes to reduce the collection size.

In this chapter, we develop a **M**ov**I**e **K**nowl**E**dge-grounded chatbot, namely MIKE, equipped with a movie knowledge base (MKB). Given an input utterance(s) associate with a topic film, our MIKE firstly recognizes the underlying attribute(s) and then collects candidate entities by starting from the mentioned entities and then propagating along the edge(s) of the recognized attribute(s). After equipped with necessary contextual knowledge, MIKE performs conversation understanding and response generation using an knowledge-enhanced Seq2Seq architecture [218]. The encoder is enhanced with the detected attributes to compress the input utterance(s) into an attribute-aware context representation. The decoder is augmented with a pointer gate [232] to decide when to mention an entity and select from the candidates the most appropriate one based on the attribute-aware context. While previous works treat attributes and entities equally, our work is novel in discerning their differences and incorporating them in different manners. In this way, our chatbot MIKE captures the conversation logic better with the help of contextual attributes, which in turn leads Mike to generate more coherent and entity-aware responses.

To validate the effectiveness of the proposed approach, we build a new movie chit-chat conversation corpus, BILI-FILM, collected from a large Chinese video platform. The contributions in this chapter are briefly summarized as follows:

- We identify external knowledge related to a conversation as contextual knowledge, and regard its necessities in both context representation and response generation.

- We propose to utilize contextual attributes and entities in their own ways. The contextual attributes are contributing to capture the conversation logic for context modeling. The related entities are beneficial to generate responses when referring is needed.

- We develop a novel movie knowledge-grounded chatbot, namely MIKE, which firstly collects contextual knowledge from a MKB we build, and then generates entity-aware responses based on the attribute-aware context representation.

- On two movie conversation corpus, our MIKE significantly outperforms other four knowledge-grounded models.

The rest of this chapter is organized as follows. Section 3.2 surveys the previous work on building knowledge-aware chatbots. Section 3.3 describes the proposed chatbot MIKE. Experiments and analysis are presented in Section 3.4. Finally, we summarize this chapter in Section 3.5.

3.2 Related Work on Knowledge-aware Chatbots

As the fundamental pillars, knowledge bases and knowledge graphs (KBs and KGs) are emerging as important data sources for various applications. Typically, a knowledge graph (KG) is a multi-relational graph composed of entities (nodes) and relations (different types of edges). Each edge is represented as a triple of the form (head entity e_h, relation r, tail entity e_t). Such a triple is also called a fact, indicating that two entities are connected by a specific relation. For example, the triple {The Notebook, actBy, Rachel McAdams} describes the fact that the film *The Notebook*

is acted by Rachel McAdams. However, the symbolic nature of KGs impedes their applications. To tackle this issue, knowledge graph embedding models have been proposed to embed the relations and entities in a KG into low-dimensional continuous vector spaces. These KG embedding models can be roughly categorized into two groups: translation-based models and semantic matching models. Specifically, translation-based models learn the embeddings by calculating the plausibility of a fact as the distance between the two entities, usually after a translation carried out by the relation. Representative models are TransE [17], TransH [246], TransR [115]. In TransE [17], the entity and relation embedding vectors are in the same space. In TransH [246], entity embedding vectors are projected into a relation-specific hyperplane. In TransR [115], entities are projected from the entity space to the relation space. [241] summarizes numerous advanced knowledge embedding approaches. In this work, we embed our KG using the widely-adopted TransE model [17], and integrate the knowledge embeddings into conversation models in a novel way.

In the line of equipping chatbots with external background knowledge, some works attempt to incorporate implicit knowledge into chatbots to address the "generic response" problem. [54] utilizes external textual information as explicit knowledge for the chatbots, and [165] exploits the relevant information within the dialogue corpus as soft prototypes to facilitate response generation. [254] presents a model allowing developers to express domain knowledge via software and action templates. Most similar to our work is [322] that develops a dialogue system to talk about musics. Although grounding on knowledge, their system is focusing more on answering music-related questions. Moreover, each dialogue in their data is restricted to one singer. Differently, our approach is targeted at film-related chit-chats on various aspects rather than answering questions in movie domains.

3.3 Method

To begin with, we describe the notation and framework of MIKE, a knowledge-grounded chatbot equipped with an associate knowledge base (KB) $\mathcal{K}$. Build upon the encoder-decoder architecture, MIKE consists of three main components, as illustrated in Figure 3.2:

- A contextual knowledge collector finds attributes and entities by linking the input sequence $\mathbf{x}$ to the associate KB $\mathcal{K}$. It detects the mentioned attributes from the input sequence, and collects entities relevant to the conversation.
- An attribute-aware encoder that transforms the input sequence of utterances $\mathbf{x}$ into an attribute-based representation by attending on the detected attributes.
- An entity-aware decoder generates the final response by properly referring to the pre-collected entities.

With these three components, our approach firstly collects from $\mathcal{K}$ the contextual knowledge pertaining to the input $\mathbf{x}$, including the related attributes and entities. The detected attributes are used in the attribute-aware encoder to form an attribute-aware context representation, while the set of related entities are used as candidates to augment the entity-aware decoder.

3.3.1 Preliminaries

In two-party human-computer conversational systems, chatbots interact with users by returning proper responses. In particular, generative conversation models formulate the problem of response generation as learning a Seq2Seq mapping.

Formally, conversation models take as input the combination of the current user utterance $\mathbf{u}^T$ and conversation histories $\{\mathbf{u}^1, \cdots, \mathbf{u}^{T-1}\}$, where T is the turn number. Each utterance in the conversation is a sequence of words, a.k.a. $\mathbf{u^t} = \{x_1, \cdots, x_{N_t}\}$.

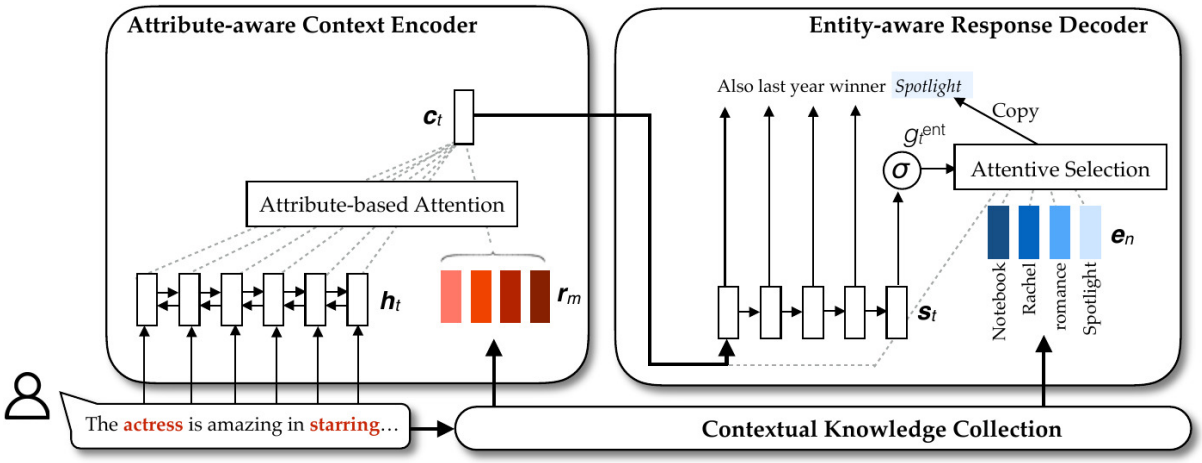

Figure 3.2: The overview of the proposed chatbot MIKE.

Hence, chatbot is fed with a sequence of words $\mathbf{x} = \{x_1, \cdots, x_{N_x}\}$, and is required to generate a response $\mathbf{y} = \{y_1, \cdots, y_{N_y}\}$, where N_x and N_y are the token numbers.

3.3.2 Contextual Knowledge Collector

The prerequisite step is to locate and extract the necessary knowledge from the equipped movie knowledge base (MKB) $\mathcal{K}$. In the left side of Figure 3.3, a general sketch of $\mathcal{K}$ is depicted in the form of knowledge graph. The nodes are entities that are connected by the attributes on the edges. Despite of the existence of countless facts in a KB, there is only a limited portion of the knowledge that are necessary for conversation understanding and response generation. It is more effective to scope a set of contextual knowledge by linking the input utterance(s) to the associate KB $\mathcal{K}$.[1]

Given a conversation, the underlying logic is often indicated by the attribute information in the utterance(s). It is feasible to detect the attribute(s) $R_{\mathbf{x}}$ from the input utterance(s) using lexical patterns because they are often expressed regularly. For example, the words "actress" and "starring" (the red bold words in Figure 3.3)

[1]For efficiency, we retrieve a subgraph of the associate topic film, and perform knowledge discovery on the subgraph.

indicate the attribute type *actBy*. Similarly, we also detect a set of entities $E_{\mathbf{x}}$ mentioned in the input utterance(s) using entity linking techniques. By including the topic film into the set $E_{\mathbf{x}}$, we produce a set of seed entities $E_{\text{seed}} = \{E_{\mathbf{x}} \cup e_{\text{topic}}\}$, where e_{topic} is the topic entity.

However, it is insufficient to solely rely on the entities explicitly mentioned in the conversation. To expand the seed set E_{seed}, we propose to collect more relevant entities E_r by using the detected attribute(s) in $R_{\mathbf{x}}$. Concretely, we take each entity in $E_{\mathbf{x}}$ as head node e_h, and collect the entity on the tail node e_t only if the relation r_{e_h,e_t} between e_h and e_t matches with (one of) the detected attribute(s) $R_{\mathbf{x}}$. In this way, only the entities linked by the detected attribute(s) are collected to expand the entity set. We repeat this procedure by 2 times, which results in a 2-hop expansion as illustrated in Figure 3.3.

Notice that it is unreliable to expand the entity set using all the attributes in the KB, although it is straightforward to do so as in [322]. The larger the size of the entity set, the harder it will be for the chatbots to reason the most suitable when generating responses. Instead, we guide the entity set expansion based on the detected attributes, which is supposed to filter out noisy entities and eventually reduce the set size. The detected attributes will bias the entity expansion to collect those entities pertinent to the inherent conversation clue, and thus encourage more smooth and coherent conversations.

As a result, we collect the set of contextual knowledge related to a conversation including the set of detected attribute(s) $R = R_{\mathbf{x}}$ and the set of candidate entities $E = \{E_{\mathbf{x}} \cup E_r\}$. To fed this knowledge into the encoder-decoder conversational model, we encode the attributes and entities into dense representation using TransE [17], a knowledge graph embedding model, and denote the resulted embeddings as $\mathbf{r}_m$ and $\mathbf{e}_n$, respectively, where $\forall m \in \{1, \cdots, N_r\}$, $\forall n \in \{1, \cdots, N_e\}$. Then, the attribute and entity embeddings are passed to the encoder and decoder in their own ways, as

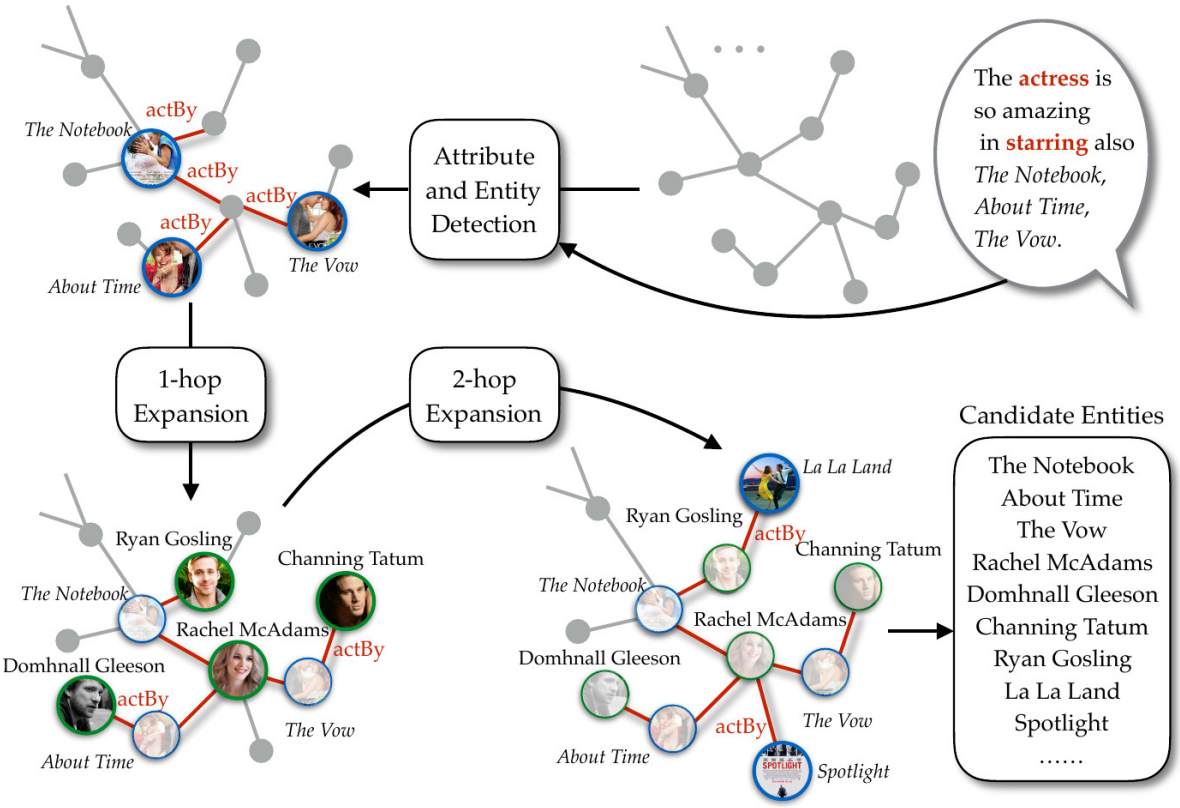

Figure 3.3: Contextual Knowledge Collector.

shown in Figure 3.2. The attribute embeddings are fed to the encoder to facilitate context modeling. The entity embeddings are served as candidates for the decoder to generate knowledgeable responses by selecting proper entities when referring is needed.

3.3.3 Attribute-aware Context Encoder

Given the input sequence $\mathbf{x}$, we embed its tokens using a Recurrent Neural Network (RNN), and then utilize the contextual attributes obtained in the first step to enhance the semantic representation.[2] This converts a sequence of inputs $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t)$ to hidden states $(\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_t)$. To capture long-term dependencies among the

[2]When $\mathbf{x}$ is a sequence of multiple utterances, we concatenate the utterances into one, unified utterance. As empirically validated, using hierarchical context encoder did not bring in obvious improvements.

utterances, we adopt a special variant of RNNs, bi-directional Gated Recurrent Unit (GRU) [32] as the encoder basis. The typical GRU cell is formed up by two gates, the update gate $\mathbf{g}_t^z$ and the reset gate $\mathbf{g}_t^r$, which are computed as follows:

$$\mathbf{g}_t^z = \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1}) \tag{3.1}$$

$$\mathbf{g}_t^r = \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1}) \tag{3.2}$$

At each time step t, the update gate $\mathbf{g}_t^z$ controls how much the unit updates the content in the hidden states, whereas the reset gate $\mathbf{g}_t^r$ acts as a similar mechanism to allow the unit forget what has been previously computed. With these two gates, the hidden states at each time step t is a linear interpolation computed as follows:

$$\mathbf{h}_t = (1 - \mathbf{g}_t^z)\mathbf{h}_{t-1} + \mathbf{g}_t^z \tilde{\mathbf{h}}_t \tag{3.3}$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_0 \mathbf{x} + \mathbf{g}_t^r) \odot (\mathbf{U}_0 \mathbf{h}_{t-1})) \tag{3.4}$$

where $\odot$ is element-wise multiplication.

To encode the semantics from both the forward and backward of the input sequences, we adopt Bi-directional GRUs as our encoder basis. The Bi-directional GRUs are two GRUs combined together. One GRU looks forward and the other one looks backward, to consume the information from both directions. As a result, the hidden states at each time step is concatenated by the state from each direction:

$$\mathbf{h}_t = [\overleftarrow{\mathbf{h}_t}, \overrightarrow{\mathbf{h}_t}] \tag{3.5}$$

Based on our preliminary studies, we propose to enrich the representation based on the detected attributes to form a context representation. As shown in the left part of Figure 3.2, we use an attribute-based attention mechanism [8] to measure the semantic relevance between the utterance hidden states and the detected attributes. We compute the attribute-attention weights as:

$$\alpha_t \sim \exp(\mathbf{h}_t \mathbf{W}_1 \bar{\mathbf{r}}) \tag{3.6}$$

$$\bar{\mathbf{r}} = \frac{1}{N_r} \sum_{m=1}^{N_r} \mathbf{r}_m$$

where $\mathbf{W}_1$ is a learned matrix. Combined with the learned attention, the final context representation:

$$\mathbf{c}_t = \alpha_t \mathbf{h}_t \tag{3.7}$$

which is then fed to the decoder. Intuitively, attribute-aware context encoder fuses the attribute information into the attribute-aware context representation, which is then used to initialize the hidden states of the decoder. When generating the responses, the attribute-aware context representation guides the decoder to prefer entities with similar representations, and thus allows the chatbot follow the underlying logic of the conversation and finally improve the response coherence.

3.3.4 Entity-aware Response Decoder

The last step is to properly respond by using the candidate entities related to the attributes. These candidate entities aid in the procedure of response generation when referring is needed.

Basically, the decoder is another GRU that takes as input the context representation $\mathbf{c}_t$ and the previously decoded token y_{t-1} to update its hidden state $\mathbf{s}_t$ similar as Eq. 3.3:

$$\mathbf{s}_t = \text{GRU}(\mathbf{s}_{t-1}, [\mathbf{c}_t; y_{t-1}]) \tag{3.8}$$

where $[;]$ is the concatenation operator of the two vectors. After obtaining the state vector at the current time step t, the decoder generates each word y_t based on a softmax classification over the hidden state $\mathbf{s}_t$ and the context $\mathbf{c}_t$:

$$\begin{aligned} p^{\text{gru}}(y_t|y_1, \cdots, y_{t-1}) &= f(y_{t-1}, \mathbf{s}_{t-1}, \mathbf{c}_t) \\ &= \text{softmax}(\mathbf{W}_o \mathbf{s}_t) \end{aligned}$$

where $\mathbf{W}_o$ is a parameter matrix. Hence, the decoder generates the response $\mathbf{y} =$

$\{y_1, \cdots, y_{N_y}\}$ conditioned on the conversation context by maximizing the probability:

$$p^{\text{gru}}(y_1, \cdots, y_{N_y}|\mathbf{c}_t) = p^{\text{gru}}(y_1|\mathbf{c}_t) \prod_{t=2}^{N_y} p^{\text{gru}}(y_t|y_1, \cdots, y_{t-1}, \mathbf{c}_t) \tag{3.9}$$

$$= p^{\text{gru}}(y_1|\mathbf{c}_t) \prod_{t=2}^{N_y} p(y_t|y_{t-1}, \mathbf{s}_{t-1}, \mathbf{c}_t) \tag{3.10}$$

To realize the entity-aware generation as illustrated in the upper right of Figure 3.2, we augment the decoder in the principle of pointer networks [232, 284, 64]. Pointer networks have been demonstrated powerful on tackling out-of-vocabulary (OOV) words during generation. Previously, they are used to copy OOV words from the input sequences into the output sequences. Inspired by this idea, we adopt pointer networks to copy entities from external KB.

Concretely, the decoder is augmented using a gate g_t^{ent}, which determines whether to copy an entity by p^{ent} or to produce a word by p^{gru}. Formally, the two-way generation is formulated as:

$$\begin{aligned} p(y_t|y_1, \cdots, y_{t-1}) = g_t^{\text{ent}} p^{\text{ent}}(y_t|y_{t-1}, \mathbf{s}_t, \mathbf{c}_t, \mathbf{E}) \\ + (1 - g_t^{\text{ent}}) p^{\text{gru}}(y_t|y_{t-1}, \mathbf{s}_{t-1}, \mathbf{c}_t) \end{aligned} \tag{3.11}$$

where $\mathbf{E}$ is the matrix stacking the candidate entity embeddings $\mathbf{e}_n$ obtained in the first step (Section 3.1). When the gate "opens", the decoder calculates the probability over the candidates $\mathbf{E}$ and then directly copies the selected entity. Otherwise, the decoder switches back to a vanilla GRU language model and omits a general word based on the softmax output. The gate g_t^{ent} is trained on the hidden state:

$$g_t^{\text{ent}} = \sigma(\mathbf{W}_g \mathbf{s}_t) \tag{3.12}$$

The remaining is to learn which entity is to be selected by p^{ent} at each time step. We adopt another attention mechanism to estimate the relevance between the context and each entity. In this way, we are able to acquire the attention weights $\boldsymbol{\beta}_t$

similar as Eq. 3.6:

$$\boldsymbol{\beta}_t \sim \exp(\mathbf{E}\mathbf{W}_e\mathbf{c}_t) \tag{3.13}$$

Since the context representation $\mathbf{c}_t$ has been enriched by the attribute embeddings, the entities connecting with the detected attributes will have similar embeddings and then attract higher attention weights. The attended entities are naturally coherent to the conversation context.

Now the augmented decoder generates a candidate entity by:

$$p^{\text{ent}}(y_t|y_{t-1}, \mathbf{s}_t, \mathbf{c}_t, \mathbf{E}) = \begin{cases} \beta_{tj}, & \text{if } y_t = e_j \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

The GRU language model p^{gru} is rather simple, and we adapt it to be aware of the film title by introducing the film title embedding in its content vector $\mathbf{c}_t$. This encourages the generation to stay focused.

3.3.5 Model Learning

After pre-collecting the candidate entities (by contextual knowledge collector), we are able to obtain supervision signals to train the switch gate g_t^{ent}. We have:

$$g_t^{\text{ent}} = \begin{cases} 1, & \text{if target word is a candidate entity} \\ 0, & \text{otherwise} \end{cases} \tag{3.15}$$

To train the model in the fully supervised manner, we have a training set of triples:

$$D = \{(X_1, Y_1, R_1, E_1)\}^{N_d}$$

where N_d denotes the count of training examples, X and Y build up the utterance-response pairs. Correspondingly, R and E are the sets of detected attributes and candidate entities, obtained using contextual knowledge collector in MIKE.

Finally, we train model parameters by minimizing the negative log-likelihood objective as follows:

$$NLL(D, \theta) = -\sum \sum_{i=1}^{N_d} \log p(Y_i|X_i, R_i, E_i) \tag{3.16}$$

The model parameters θ include the embeddings of vocabulary, entities, relations, and the encoder-decoder components. Since the model is fully differential, we use stochastic gradient descent to back-propagate the gradients through the model components.

3.4 Experiments

In this section, we acquire two movie conversation corpus, on which we compare with 7 state-of-the-art conversational models to demonstrate the effectiveness of the proposed approach. As indicated by the automatic evaluations and human judgments, the proposed MIKE outperforms other knowledge-grounded models significantly.

3.4.1 Datasets

We evaluate the proposed chatbot using two movie conversation corpus. The first corpus we adopt is a publicly available knowledge-driven dialog dataset, DUCONV,[3] a carefully-crowdsourced conversation dataset. In DUCONV, each dialog is formed by two human crowdsourcers, where one human plays the role of leading the conversation, i.e., given related knowledge, initiating a novel topic or continuing the current one in the movie domain [260]. The DUCONV dataset consists of 30k conversations with 120k conversation turns. We randomly split the dataset by 8:1:1 into training/development/test set.

[3] https://github.com/PaddlePaddle/models/tree/develop/PaddleNLP/Research/ACL2019-DuConv

Table 3.1: Statistics of Corpus DuConv and Bili-film.

Dataset	DuConv	Bili-film
Total Number of Conversations	29,858	12,530
Total Number of Utterance	270,399	38,467
Average Number of Speaker Turns	9.1	3.6
Average Number of Tokens Per Turn	10.6	27.8
Number of Covered Movies	91,874	187
Number of Covered Movie Stars	51,753	248
Number of Unique Entities Per Conversation	9.3	3.1

In addition to this carefully-curated corpus, we also build a novel conversation corpus Bili-Film from real-world data. Bili-Film is crawled and curated from a Chinese video discussion platform Bilibili.[4] Although there are other movie discussion platforms,[5] the discussions on them are often focusing on detailed plots, and are too complex to learn. In contrast, the discussions BiliBili are more condense to capture.

The BiliBili users often release movie-related videos such as self-produced lens, montages, and narrations. Other users may discuss on the videos by introducing comments or replying to other users under the videos, which is as usual as on typical forums. The comment threads between two users are the desired discussions we collect. We collect a set of 20 seed active publishers to crawl the two-party discussions under their videos. We also extract the corresponding film titles from the video captions, and use them as the discussion topics. We filter out some discussions that are meaningless or too long to learn. We maintain at most four speaker turns in all discussions. After random spilt, 10,000 conversations in Bili-film are used for training, 1,530 for validation, and 1000 for testing.[6] The statistics of DuConv and

[4] `https://www.bilibili.com/v/cinephile/`

[5] i.e., `https://www.reddit.com/r/movies/`, `https://moviechat.org/`, `https://filmboards.com/`, etc.

[6] The dataset will be released to the public.

our BILI-FILM corpus are presented in Table 3.1.

3.4.2 Experimental Setup

Knowledge Base Construction

To build a KB from scratch requires tedious effort. Instead, we build our KB $\mathcal{K}$ by leveraging the largest Chinese KB ZHISHI.ME [155], which comprises a lot of knowledge from three encyclopedias: Baidu Baike, Hudong Baike, and Wikipedia in Chinese.[7] Even though it is a general KB, ZHISHI.ME is focused on movie domain an covers more facts than those in CN-DBPedia [270].

In specific, we firstly extract the triples from ZHISHI.ME whose attribute types are either *actBy* or *directBy*. This is assumed to collect all the film entities in ZHISHI.ME. We also adopt a common practice that adding the inverse attributes, i.e., $actBy^{-1}$) to cover more facts. After that, we run a second round of collection focusing on the triples whose attribute types are: *actBy*, *writeBy*, *directBy*, *hasAlias*, and *hasGenre*. The reason why we adopt *writeBy* is to include the cases when the movies are adapted from books like the series of Harry Potter. As a result, our KB is defined with five entity types, i.e., actor (actress), writer, director, film, and genre. Another thing to note is that, the entities are either real-world concrete things, e.g., *Harry Potter*, or virtual concepts, e.g., comedy movie. For simplicity, we do not distinguish them undistinguished in this work.

As mentioned before, entity alias mining is crucial in our scenario. To improve the performance of entity discovery, we refine our MKB by extracting more alias information from an extra source. Although entities in ZHISHI.ME already contain the attribute *hasAlias*, they are sometimes out-of-date. To cover more, we also acquire alias from Douban Movie. For example, the famous Chinese director Stephen Chow (周星驰) are mostly mentioned with his nicknames 周星星 and 星爷. However,

[7] `https://baike.baidu.com`, `https://www.hudong.com`

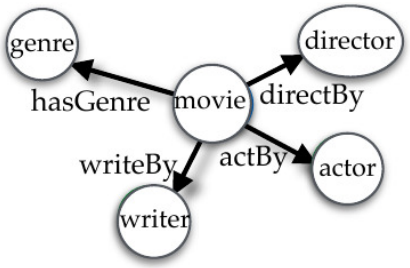

Figure 3.4: The Schema of the MKB.

the former one is missing in ZHISHI.ME but found in Douban.[8] These additional nicknames are appended to facilitate entity linking.

Compared Models

In order to examine whether MIKE is effective on incorporating knowledge into conversation modeling, we validate the performances of the following approaches:

- ATTN-ENC-DEC [8]: It is a standard encoder-decoder approach with the widely-adopted attention mechanism. The encoder and decoder in this models are set as GRUs [33] for fair comparison. Note that neither history utterances, nor extra knowledge is incorporated. We choose this bare-bones model to demonstrate to what extend the performance will be achieved by a standard Seq2Seq conversational agents without knowledge.

- CONCAT-ENC-DEC [209]: This model is extended from ATTN-ENC-DEC where history utterances are concatenated along with the current input, and still without background knowledge.

- HRED [208]: This state-of-the-art model incorporates history utterances, where a conversation-level ContextRNN is on the top the word-level utteranceRNN.

[8] `https://movie.douban.com/celebrity/1048026/`

- FACT-ENC-DEC [54]: It is a knowledge-grounded conversation model that consumes relative textual facts as additional knowledge information. To fit it into our scenario, we use the films' one-sentence descriptions as the textual facts. By comparing with it, we aim at distinguishing the effects between the unstructured and structured knowledge.

- KB-LSTM [282]: It identifies the knowledge related to the conversation and encodes the knowledge into conversation representation, which is similar with our idea. Differently, KB-LSTM only encodes the entities explicitly mentioned in the input utterance, and incorporates the entity encodings using concatenation operation in the encoder. On the contrary, we feed the context-relevant entities to the decoder for reasoning in response generation while our encoder takes the attribute information into account.

- KB-LSTM+: We improve the above KB-LSTM model by incorporating also attributes information into the corresponding encoder. This is assumed to inject more knowledge implicitly and thus expand its knowledge scope. We denote this enhanced version as KB-LSTM+.

- GENDS [322]: It is the most similar approach to ours. GENDS shares a similar idea with ours that it ranks candidate entities collected from the retrieved facts to facilitate entity-aware response generation. Because candidates in GENDS also contain the entities implicitly mentioned in the input, it mainly differs with ours in how the candidate entities are selected.

All models are implemented by TensorFlow [1]. For pre-processing, we the utterances are tokenized using Jieba segmenter.[9] To encode the movie knowledge base, we apply the KB2E [116] implementation of TransE graph embedding approach.[10] The

[9] `https://github.com/fxsjy/jieba`

[10] `https://github.com/thunlp/KB2E`

vocabulary size is constrained to 25,000 words, and the words are initialized using 300-dimentional FastText vectors [15]. During training, these word embeddings are then fine-tuned. The sizes of the hidden states are all set as 512, and the size of mini-batch is 32. We set the initial learning rate as 0.001. During learning, it is adapted it to be exponentially decayed, and the learning gradients whose norms are larger than 0.5 are also clipped. We adopt the Adam optimizer for training stability [91].

Evaluation Metrics

In order to assess the model performances, we use a set of popular metrics that are commonly adopted in previous works to evaluate response quality [218, 127, 124, 195, 200], including both automatic evaluations and human judgments:

- BLEU-n: The N-gram based BLEU scores are proposed to indicate the overlapping degree between the generated responses and the ground-truth response [162];
- Dist-n: Since the distinct grams produced by the models stand for the informativeness of the responses, it is reasonable to devise a measurement based on it to evaluate response quality. Typically, the Dist-1 and Dist-2 scores stand for the ratios for unigrams and bigrams [101]. This metric has been widely used in works on response generation [268, 265];
- Appropriateness and Grammar: According to previous studies [124], the aforementioned automatic metrics do not often correlate well with human judgments in conversation generation tasks. To ease this issue, we also evaluate the models using human judgments. We first adopt two 3-scale human evaluation metrics, Appropriateness and Grammar, to judge the quality of the generated responses [195];
- Precision and Recall: These two scores are used to examine the overlapping on knowledge-specific words, i.e., entity mentions in the generated responses [322].

The precision is the percentage of right generated entities in all generated entities, whereas the recall is of ground truth entities. These two metrics are calculated based on 100 manually annotated cases and are used to examine the ability of referring to the most relative entities. In this case, generating responses that contain irrelevant entities are not preferred.

3.4.3 Performance Evaluation

As shown in Table 3.2, the experimental results on the two corpus **DuConv** and **Bili-film** exhibit similar findings. We first examine the importance of knowledge in chat response generation. As implied by the first three rows across the two datasets, the three chatbots in each first block perform the worst. This is not surprising, because they are models with no access to contextual knowledge. Such performances are disappointing but reasonable, and motivate our research to incorporate background knowledge into open-domain dialogue models.

Since all other five models are equipped with background knowledge, we then investigate among them which mechanism(s) is more effective in utilizing background knowledge. According to the form of knowledge they consume, these five models can be further categorized into two groups: unstructural knowledge v.s. structural knowledge. It is obvious that Fact-Enc-Dec lags far from other knowledge-grounded models on DuConv. Even worse, Fact-Enc-Dec performs almost similar as Attn-Enc-Dec even it consumes extra knowledge on Bili-film. Notice that the fact knowledge it utilizes is represented in the form of natural language sentence, i.e., *Titanic stars Leonardo DiCaprio and Kate Winslet as....* Such unstructured representation impedes existing encoder-decoder models to exploit useful information from it and results in negligible improvement over the "non-knowledge-aware" Attn-Enc-Dec. On the contrary, KB-Lstm, KB-Lstm+, GenDS and our Mike utilize structural knowledge, i.e., the attributes and entities. Their better

Table 3.2: Model Comparison Results on DuConv and Bili-flim.

DuConv								
	Automatic Evaluations				Human Judgments			
Model	**BLEU-2**	**BLEU-3**	**Dist-1**	**Dist-2**	**Appr.**	**Gram.**	**Prec.**	**Recall**
Attn-Enc-Dec	0.12	0.08	0.03	0.06	1.64	1.88	0.20	0.07
Concat-Enc-Dec	0.11	0.08	0.02	0.06	1.56	1.81	0.16	0.07
HRED	0.14	0.09	0.10	0.05	1.68	1.80	0.19	0.10
Fact-Enc-Dec	0.26	0.15	0.13	0.11	1.66	1.83	0.18	0.08
KB-Lstm	0.37	**0.29**	0.15	0.16	1.72	**1.84**	0.30	0.17
KB-Lstm+	0.34	0.23	0.14	0.12	1.68	1.83	0.26	0.15
GenDS	0.38	0.20	0.14	0.08	1.70	1.79	0.34	0.22
Mike	**0.44**	**0.29**	**0.20**	**0.18**	**1.84**	1.82	**0.38**	**0.25**
Bili-film								
	Automatic Evaluations				Human Judgments			
Model	**BLEU-2**	**BLEU-3**	**Dist-1**	**Dist-2**	**Appr.**	**Gram.**	**Prec.**	**Recall**
Attn-Enc-Dec	0.73	0.18	0.03	0.08	1.55	1.79	0.14	0.14
Concat-Enc-Dec	0.76	0.20	0.03	0.10	1.58	1.72	0.14	0.15
HRED	0.68	0.17	0.02	0.11	1.72	1.34	0.13	0.14
Fact-Enc-Dec	0.82	0.19	0.06	0.16	1.75	1.80	0.13	0.08
KB-Lstm	1.13	0.32	0.11	0.19	1.82	1.86	0.31	0.23
KB-Lstm+	1.13	0.37	0.14	0.25	1.82	2.00	0.32	0.31
GenDS	1.09	0.68	0.16	**0.43**	1.96	2.03	0.37	0.38
Mike	**1.27**	**0.84**	**0.19**	0.40	**2.40**	**2.15**	**0.47**	**0.53**

performances on both DuConv and Bili-film suggest that it is more effective to integrate knowledge in structured forms into Seq2Seq models.

Among the group of models that utilize structural knowledge, KB-Lstm, KB-Lstm+ lag far from Mike especially on BLEU-n, Distinct-n and entity-related scores. While both KB-Lstm+ and KB-Lstm+ employ attribute and entity information, KB-Lstm+ results in negligible improvement (and even decrease) over the original KB-Lstm. The difference between KB-Lstm+ and Mike lies largely in the distinct mechanisms they utilize the attribute and entity information. KB-Lstm+ comprises the attribute and entity information into a single vector and passes it to the RNN hidden state, which might be too elusive to guide high-quality response generation. Differently, Mike makes use of attributes and entities in different manners. Mike attends on the detected attributes to fuse the attribute information into context representation, and mentions the proper candidate entity(s) whenever the pointer gate is activated. The comparison results demonstrate the superiority of Mike on incorporating this knowledge.

Overall, according to their performances on both DuConv and Bili-film, Mike and GenDS perform the best and the second best, accordingly. Their similarity is the entity reasoning ability which learns to select entities from the pre-collected set of candidates. This proves that such a mechanism is necessary for chatbots. Different from Mike, GenDS retrieves entities by string matching the fact triples in the KB with the entities explicitly mentioned in the conversation utterances. In such unfiltered way, their candidate set might include noisy entities that are too tangential to the conversation context. As a result, GenDS has larger possibilities of attending on wrong, peripheral entities, and then generates unintelligible responses. On the contrary, Mike accesses to new entities E_r linked by the detected attributes. The detected attributes will bias the entity expansion to collect implicit but material entities that closely related to the conversation. This novel strategy enables

MIKE to expand the conversation scope, and meanwhile limits the candidate set in a reasonable range.

On the two corpus DUCONV and BILI-FILM, the proposed MIKE surpasses the compared models significantly in terms of almost all indicators. Notably, the automatic Distinct-n scores and human evaluation scores (Appr., Gram., Prec., and Recall.) indicate that the responses generated by our MIKE are more diverse, fluent, and appropriate to the conversation context.

3.4.4 Analysis

MIKE consists of three modules, i.e., contextual knowledge collector, attribute-aware encoder and entity-aware decoder. To examine the performance and contribution of each module, we conduct ablation studies and error analysis on BILI-FILM. We randomly select 100 test cases, and manually annotate the attribute and entities in the input utterance.

Attribute and Entity Detection

Note that in our case, the underlying attributes are often expressed regularly. Most entities mention in the text are movie-related. More importantly, we only care about those attributes and entities related to a specific given film. Hence, we use simple matching algorithms to separately detect attributes and entities from text.

Given an input, the attributes are detected automatically by lexical patterns. For example, the appearance of "actress", "starring in", "has a role of" indicate the attribute *actBy*. Based on the identified attributes, entity mentions are detected through string matching. Although some APIs are able to extract entities from short text, we find they are unreliable since the recall of a 100 test example are less than 10%. More advanced approaches as in [282] might be of help but we leave it as future work. To improve matching quality, we clean the punctuations in advance.,

Table 3.3: Error Analysis of Attribute and Entity Detection.

Attribute			**Entity**		
Correct	94	76.4%	**Correct**	128	64.7%
Missing	12	9.8%	**Missing**	65	32.8%
Wrong	17	13.8%	**Wrong**	5	2.5%

i.e., guillemets (《》), interpuncts (·) and quotation marks (“”). As a result, 莱昂纳多·迪卡普里奥 (Leonardo DiCaprio)=莱昂纳多-迪卡普里奥=莱昂纳多迪卡普里奥. To accelerate, we also segment the entity names and match them in segment units. In this case, “Leonardo” will also be successfully matched to “Leonardo DiCaprio”.

The performance of contextual knowledge collection is reported in Table 3.3. Since there often exist multiple attributes and entities in each utterance, the total number of the annotated ones are more than 100. It is shown that our detection accuracies are 76.4% and 64.7%, which are comparable to the performances in similar settings [298].

Although our scenario is much simpler, pattern matching techniques still face challenges. We show some cases in Table 3.4. As shown in Case #1, simple pattern matching will fail when the sentence has negative terms. This indicates that semantic parsing is needed when complex sentence grammar like concessive clause exists. Sometimes, the indicator word (pattern) is misleading as in Case #2. Another kind of failure is caused by entity detection. In Case #3, the model fails to link the mention “Old Leo” to the entity *Leonardo* because the associate KB does not cover the alias “Old Leo”. Note that the last case is about the conversation on the film *Leon: The Professional* directed by Luc Besson. However, the user mentions the director Stephen Chow, which is not covered in the subgraph of *Leon*. Theoretically, it is applicable to link entities based on the whole graph, which we leave as future work.

Table 3.4: Case Study of Attribute and Entity Detection.

Case	Input Utterances	Detected	Truth
1	我觉得就是因为星爷不是**主演** (I think it is because that Stephen Chow is not the leading **actor**.)	actBy	directBy
2	电影的**演员**是砖瓦，而特效仅仅是房子的装饰 (For a film, the **actors** are tiles, while special effects are only decorations.)	actBy directBy	directBy
3	老李子那个拿杯酒嘴角微翘的笑容堪称影片灵魂 (The shot that "Old Leo" holds the glass and smiles is the soul of the film.)	None	Leonardo
4	这个电影让我想起了周星驰的回魂夜 (The film reminds me of Stephen Chow's *Out of the Dark*.)	None	Stephen Chow Out of the Dark

Table 3.5: Ablation Studies.

Model	BLEU-3	Distinct-1	Precision	Recall
MIKE	0.84	0.19	0.47	0.53
-2HE	0.70	0.13	0.31	0.38
-AAE	0.55	0.11	0.24	0.24
-EAD	0.19	0.04	0.16	0.08

Ablation Studies

We perform additional ablation studies to investigate how important the following parts in our approach are: (1) the "2-hop-expansion" (2HE) solution in candidate entity selection; (2) the attribute-aware encoder (AAE); (3) and the entity-aware decoder (EAD). Table 3.5 presents the experimental results. For comparison purpose, we list the performance scores achieved by our full model MIKE in the first row.

After removing the 2HE trick, as shown in the second row, the precisions and recalls will drop to 0.31 and 0.38 respectively, which indicates that it is necessary to expand the conversation scope by enlarging the candidate entities. As shown in Figure 3.3, to allow richer and more diverse semantics in the conversation, we treat the detected entities as seeds and add their neighboring entities that are linked by the detected attributes within 2-hops. Since we include all the detected and re-collected entities as the candidates, the proposed 2HE trick is beneficial to generate more diverse and informative responses.

After replacing the attribute-aware encoder with a vanilla RNN encoder, the performance scores also decreases. This suggests that attribute-aware encoder is also crucial to facilitate conversation understanding by using the contextual attribute information. Intuitively, the attribute-aware context encoder fuses the attribute information into the attribute-aware context representation, which allows the chatbot follow the underlying logic of the conversation when generating the responses.

Our approach degrades to standard ENC-DEC when all the special designs are

removed. The remarkable gap between the scores in the last two rows are strong evidence for the necessity of the entity-aware decoder. Essentially, the decoder in the proposed MIKE is a RNN language model augmented with the pointer gate [232, 284, 64]. In this way, it ranks candidate entities collected from the associate knowledge base and thus generates more engaging and informative responses.

3.5 Chapter Summary

In this chapter, we investigate conversation modeling using external knowledge, and propose a knowledge-grounded conversational model called MIKE. Based on the encoder-decoder architecture, the proposed MIKE consists of three main components: (1) a contextual knowledge collector that performs knowledge discovery and transfer to link the associate KB with the given conversation; (2) a novel attribute-aware context encoder that represents current and history utterances using the collected attribute information; (3) a powerful entity-aware response decoder that generates informative responses by properly referring to suitable entities. With these three components, the proposed MIKE are able to comprehend conversation logic using the detected attributes and respond to users more engagingly and coherently using the candidate entities.

On two movie conversation corpus DUCONV and BILI-FILM, we empirically demonstrate the effectiveness of MIKE. It significantly outperforms other 7 state-of-the-art conversation models through both automatic evaluations and human judgments. The generated responses by MIKE are the most plausible among the compared ones. We further conduct error analysis and ablation studies, and investigate the importance of each component in our approach. The overall experimental results reveal that attribute and entity information play distinguished and indispensable roles in conversation modeling, which have been neglected in previous research.

Chapter 4

Emotion Incorporation for Utterance-level Coherence

4.1 Introduction

In order to build high-quality conversational agents, some researchers have investigated the human expectations over the virtual conversational agents. Many previous studies pointed out the importance of humanness towards building a natural conversational agent [21]. The most frequently mentioned criteria are responding coherently with the preceding context [129, 140, 85, 154], learning user needs and answering questions [129, 140], realizing domain-specific concepts and terms [296, 154], facilitating input and response diversity [129, 140, 85, 151], and developing a consistent personality [140, 85, 154].

In addition, emotional intelligence has also been emphasized, which is an important human intelligence. Indeed, emotions play a vital role in our daily lives and fundamentally affect people's communication. When friends express their upset moods, people usually sympathize with them and ask why [139]. Research shows that using emotional intelligence to recognize conversational agents can increase user engagement and satisfaction [167, 199, 98]. Psychological or mental states such as empathy contribute a lot to emotional intelligence. In accordance, these mental

A: **Would you like to go dance with me tonight? (happy)**
B: *Sounds great! Come by any time!* (happy)
A: Cool. Let's meet at the club on 8 p.m.
B: No problem, see you then. (happy)

B: *I am just not in the mood for this.* (sad)
A: You look so upset. What's going on?
B: I lost the table tennis game yesterday. (sad)

Figure 4.1: A Motivating Conversation Example.

states then stimulate the relevant behaviors in human's daily chat.

Therefore, we take emotion as a representative kind of information to investigate its importance in open-domain chatbots. To endow the chatbots with emotional intelligence, there are at least two issues to resolve:

- Take the conversation in Fig 4.1 for example. As we can see, person B receives a dance invitation from A, but shows different wishes under certain emotions. In the upper case, the conversation continues in a happy way and arrives at a dance date when the conversation ends. When comes to the lower case, friend B is frustrated because of the failure of the game, so he/she declines the dance invitation. This example shows that emotions often guide people's words and thus the outcome of dialogue. Therefore, dialogue agents with emotional intelligence should incorporate emotional information when generating responses.

- However, emotion is an internal speaker state in the process of conversation, which cannot be directly obtained in actual application scenarios. In this example, when friend A is aware of friend B's grief, he/she expresses concern about A. Therefore, it is very intuitive for the chatbot to perceive the user's emotions. Therefore, intelligent agents are expected to detect users' emotions and reply emphatically using more pleasing expressions.

In this chapter, we handle these two issues by proposing a conditional variational

framework, enabling controlled response generation by a specific attribute, e.g., emotion. Inspired by the semi-supervised deep generative model [92], our framework produces responses regarding to not only the conversation context, but also a stochastic variable as well as an external label. In addition, we also propose to keep two separate dialogue contexts for each speaker in the conversation, in order to learn the speaker-aware information like personality, sentiment, styles, etc. To resolve the two issues discussed above, we test our framework on a scenario where the label serves as a signal to indicate which kind of emotion should the response embodies. Before the decoder starts the generation, the proper emotion class is inferred in advance to lead the response semantics. We name the framework as SPHRED.

To validate the effectiveness of the proposed framework, we curate a novel dataset named as DailyDialog [113], which consists of 13,118 dialogues with high-quality manually labelled emotion information. In brief, we highlight our contributions as follows:

- We propose a conditional variational framework for emotion-aware dialog generation, and provides the context vector for both speakers separately.
- We curate a high-quality conversation dataset DailyDialog, which consists of manually-labeled emotion information. Our dataset is released to the public on `http://yanran.li/dailydialog`, and has been included in the popular huggingface NLP platform as an benchmark conversation dataset.[1] We believe it will fortune the research future r the furtureworks in esearch in this field.
- We verify the proposed framework on emotion incorporation. The evaluation results on DailyDialog show that SPHRED providess a better context representation than previous models and helps generate higher-quality emotion-aware responses.

[1] `https://github.com/huggingface/datasets/pull/556`

The remaining of this chapter is organized as follows. In Section 4.2, we conduct survey of existing emotion-aware chatbots. Then, we describe our method on modeling emotion information for response generation in Section 4.3. In Section 4.4, we present the proposed dataset as well as the evaluation results and analysis, followed by our chapter summary in Section 4.5.

4.2 Related Work on Emotion-aware Chatbots

Affective computing, initiated by Picard [169] in the mid-1990s, is expected to be an essential ability of computer and is thus argued as a critical direction of human-computer interaction research. For example, earlier works show it is helpful for users to get emotional support and overcome frustration if computer and machines are able to perform communication strategies like active listening and exhibit the sense of empathy during the interactions [96]. Another study [13] establishs that even after a long course of interaction, users find an embodied relational agent with deliberate social-emotional skills are more respectful, appealing, and trustworthy than an equivalent task-oriented agent.

In this chapter, we mainly focus on incorporating the factor of emotion, which is an essential aspect of building human-like chatbot. The rise of the emotion-aware chatbot (EAC) Parry [34] is inspired by ELIZA [247] in early 1975. In early development, EAC is designed by using a rule-based approach. Similar to Eliza, Parry still uses a rule-based approach but with a better understanding, including the mental model that can stimulate emotion. Now, most of emotion-aware chatbots are built upon neural network models. In May 2014, Microsoft introduced XiaoIce [200], an empathetic social chatbot which is able to recognize users' emotional needs. XiaoIce is able to provide an engaging interpersonal communication by giving encouragement or other affective messages, so that succeeds in holding human attention during com-

munication.

There are at least two vital parts of building EAC, an emotion classifier to detect emotion contained in the utterances, as well as an emotion-aware response generator to produce a emotional and meaningful response. Emotion detection is a well-established task in natural language processing research area. In the early development of emotion classifier, most of the studies propose to use traditional machine-learning approach. In recent years, neural network based approachs are able to gain better performances. Based on the detected emotion categories, the chatbots will respond with the most appropriate emotion. One recent emotion-aware chatbot in the research area is Emotional Chatting Machine (ECM) [312]. Then several studies follow to deal with this research area by introducing emotion embedding representation [6, 192, 36] or modeling as reinforcement learning problem [214, 100].

There are also an expanding numbers of emotion-aware chatbots, which are developed using other kind of approaches. Several studies design emotion-coping approaches by adjusting the neural network structure and the training objective function to make the model produce responses following a predefined strategy [5, 309, 267]. Other bodies of work employ explicit indicators, such as the use of emoji, image, or emotional category, to inform their model how to regulate the emotional response [317, 76, 83, 206]. Our work is inspired by the idea of modeling discourse-level latent variable, and we attempt to incorporate interpretable variables into these Seq2Seq models like emotion variable.

4.3 Method

4.3.1 Preliminaries

Varitional Autoencoder

For general VAE [94, 182], it is a deep generative latent model that simultaneously learns an encoder and decoder from a set of data samples. Typically, VAE assumes that a continuous variable $\mathbf{z}$ is generated from a pre-defined prior distribution $p_\theta(\mathbf{z})$ and the data is then produced from the condition distribution $p_\theta(\mathrm{x—z})$, where $\mathbf{x}$ represents the data samples. Commonly, Gaussian density is adopted but other choices are possible, such as Bernoulli or Poisson.

On one hand, the encoder in the VAE architecture approximates the distribution $q_\phi(\mathbf{z}|\mathbf{x})$, which captures a hidden representation of data samples $\mathbf{x}$. On the other hand, the VAE's decoder attempts to capture the distribution $p_\theta(\mathbf{x}|\mathbf{z})$, which enables transforming hidden representations into an output. As such, a general VAE model is aimed to optimize the following objective:

$$-\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}[\log(p(\mathbf{x}|\mathbf{z})] + KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{4.1}$$

The first term in Eq. 4.1 is the expected negative log likelihood of the data distribution, which is often called the *reconstruction loss*. Intuitively, such loss encourages the encoder to better capture the underlying distribution of the data samples. The other term in Eq. 4.1 is the Kullback-Leibler divergence between the encoder's distribution $q_\phi(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z})$, which calculates the information discrepancy when approximating $\mathbf{z}$ using $q_\phi(\dot{)}$ and encourages the true latent distribution to be aligned with the (pre-defined) Gaussian distribution.

Conditional Varitional Autoencoder

Typically, the conditional variational autoencoder (CVAE) is a conditional variant of general variational autoencoder (VAE), which introduces a probabilistic distribution

over the latent variable to model response diversity. Following CVAE, we firstly encode $\mathbf{x}$ and $\mathbf{y}$ by the post encoder and response encoder, respectively. The two encoders are constructed by the shared bidirectional GRUs [33] which generate a series of hidden states $\{h_{x_i}\}_{i=1}^{|\mathbf{x}|}$ for $\mathbf{x}$ and $\{h_{y_i}\}_{i=1}^{|\mathbf{y}|}$ for $\mathbf{y}$. Then, we obtain the sentence representation $\overline{h_x}$ for the post $\mathbf{x}$ by averaging $\{h_{x_i}\}_{i=1}^{|\mathbf{x}|}$. The sentence representation $\overline{h_y}$ for the response $\mathbf{y}$ is calculated from $\{h_{y_i}\}_{i=1}^{|\mathbf{y}|}$ in the same way.

In training phase, we sample a latent variable z from the posterior distribution $q_R(z|\mathbf{x}, \mathbf{y})$. The distribution is modeled as a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, where Σ is a diagonal covariance. We parameterize μ and Σ by the recognition network through a fully connected layer conditioned on the concatenation $[\overline{h_x}; \overline{h_y}]$:

$$\begin{bmatrix} \mu \\ \log(\Sigma) \end{bmatrix} = W_q \begin{bmatrix} \overline{h_x} \\ \overline{h_y} \end{bmatrix} + b_q \tag{4.2}$$

where W_q and b_q are learnable parameters. To mitigate the gap in encoding of latent variables between train and testing [203, 280], CVAE requires the posterior distribution $q_R(z|\mathbf{x}, \mathbf{y})$ to be close to the prior distribution $p_P(z|\mathbf{x})$. Notably, $p_P(z|\mathbf{x})$ is parameterized by the prior network and also follows a multivariate Gaussian distribution $\mathcal{N}(\mu', \Sigma')$ in a similar way but only conditioned on $\overline{h_x}$. As usual, we minimize the discrepancy between the two distributions by the Kullback-Leibler divergence in part to the total marginal lower bound:

$$\mathcal{L}_{kl} = KL(q_R(z|\mathbf{x}, \mathbf{y})||p_P(z|\mathbf{x})) \tag{4.3}$$

Varitional Encoder Decoder

Although it is simple, it is not enough to apply VAE to NLP tasks like text summarization and dialogue response generation. These tasks require the models to map

and "translate" the input $\mathbf{X}$ into an output $\mathbf{Y}$ with different semantics. To remedy this issue, researchers devise VAEs with the framework of encoder-decoder, and the resulted variational encoder-decoder (VED) framework is equipped with an an extra inference network on $\mathbf{X}$, i.e., $q_\phi(\mathbf{z}|\mathbf{y}) = q_\phi(\mathbf{z}|\mathbf{Y}(\mathbf{x})) = q_\phi(\mathbf{z}|\mathbf{x})$.

Whereas classic Seq2Seq models represent each example $\mathbf{x}$ in the corpus $\mathbf{X}$ using a fixed representation, VEDs inject the stochastic variable $\mathbf{z}$ into the decoders by modifying $p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})$. Therefore, sampling z for several times will vary greatly the generation process even when the input $\mathbf{x}$s are the same. The proposed method in this chapter also adopts VED as the basic architecture, which takes history utterances as input and uses a variable encoder-decoder architecture [189] to transform the input. In specific, a low-level encoder consumes the current utterance and a high-level encoder is used to represent the history dialogues. In [189], these two encoders are named as EncoderRNN and ContextRNN for utterance-level and context-level computation as:

$$\mathbf{u}_t = \text{EncoderRNN}(x_1, \cdots, x_{t-1}) \tag{4.4}$$

$$\mathbf{c}_t = \text{ContextRNN}(u_1, \cdots, u_t) \tag{4.5}$$

$$\mathbf{c}_t = f(\mathbf{u}_{t-1}, \mathbf{c}_{t-1}) \tag{4.6}$$

where $\mathbf{u}_t$ and $\mathbf{c}_t$ are the hidden states of the low-level EncoderRNN and high-level ContextRNN, respectively. During the course of the conversation, ContextRNN summarizes the historical information gradually and finally form the representation $\mathbf{c}_t$.

When generating responses, a DecoderRNN parametrized by p_θ produces the subsequent response word-by-word via:

$$y_t \sim p_\theta(y_t|\mathbf{c}_t, y_1, \cdots y_{t-1})$$

As analyzed before, it is better to model more diversity during the generation.

To capture variation during decoding, there is a latent variable $\mathbf{z}_c$ injected into the DecoderRNN in order to influence the generation by:

$$y_t \sim p_\theta(y_t|\mathbf{z}_c, \mathbf{c}_t, y_1, \cdots y_{n-1}) \tag{4.7}$$

Initially, $\mathbf{z}_c$ was brought in for language modeling and one-sided sentence generation [20], which has been used to capture high-level information like themes, emotions, styles, and other interpretable features [20, 189]. However, the ambiguity in $\mathbf{z}_c$ makes it difficult to capture precisely. In addition, if other influencing factors can be captured from the conversation utterances, it will complement the data insufficiency and help mitigate the ambiguity in the response semantics to be uttered.

4.3.2 Separated Context Modeling

To better capture the conversation context, which is shown criticial for response generation, we develop a hierarchical framework with separated context modeling (SPHRED). This section firstly introduces the concept of SPHRED, then describes in detail the conditional variational framework and how to apply the framework to build emotion-aware chatbots.

We decompose a conversation as a two-level hierachical sequences: sequences of utterances in the top-level and sequences of tokens in the word-level, as in [208]. Let $\mathbf{w}_1, \ldots, \mathbf{w}_N$ be a conversation with N utterances, where $\mathbf{w}_n = (w_{n,1}, \ldots, w_{n,M_n})$ is the n-th speaker turn. The probability of the conversation sequences can be factorized as:

$$\prod_{n=1}^{N} \prod_{m=1}^{M_n} P_\theta(\mathbf{w}_{m,n}|\mathbf{w}_{m,<n}, \mathbf{w}_{<n}) \tag{4.8}$$

where θ includes the model parameters and $\mathbf{w}_{<n}$ represents the conversation context up to the step n.

If we only use a single RNN to model the conversation context, it will result in a general context representation, which fails to learn the distinguished speaker-

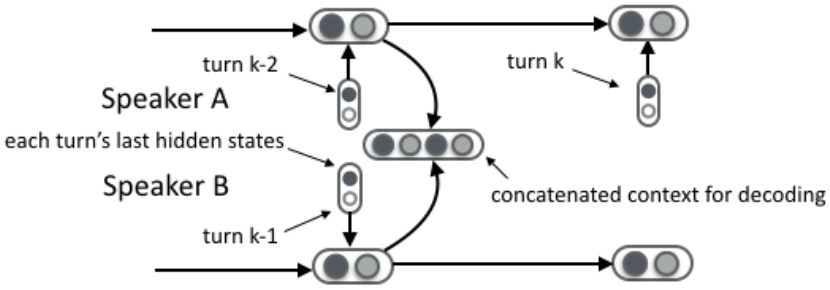

Figure 4.2: Computational Graph for SPHRED.

aware information for each speaker. It is unsuitable for us to adopt such common representations especially when we are interested in exploiting the personal attributes that can be acquired from the model and applied to guide the variable learning. Hence, we instead build two separate states for the two speakers, which is illustrated in Figure 4.2. Concereteley, the proposed SPHRED consists of an token-level encoder RNN, along with two speaker-aware status RNNs, each assigned for a corresponding speaker. When processing the speaker turn k, each status RNN takes as input the last encoder RNN state of turn $k-2$. Then, the obtained two status vectors are concatenated to form the higher-level context representation.

We will demonstrate in the experiments that the developed SPHRED not only better captures speaker-aware information using each individual status RNN, but also learns a meaningful context representation than the original HRED [208].

4.3.3 SPHRED

VAEs have been used for text generation in [20], where texts are synthesized from latent variables. Starting from this idea, we assume every utterance $\mathbf{w}_n$ comes with a corresponding label $\mathbf{y}_n$ and latent variable $\mathbf{z}_n$. The generation of $\mathbf{z}_n$ and $\mathbf{w}_n$ are conditioned on the dialog context provided by SPHRED, and the additional class label $\mathbf{y}_n$. For each utterance, the latent variable $\mathbf{z}_n$ is first sampled from a prior

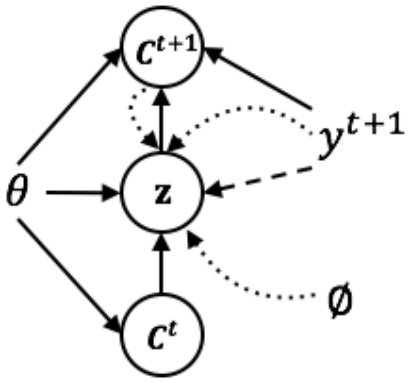

Figure 4.3: Graphical Model for the Conditional Variational Framework.

distribution. The overall process of dialogue generation can be formulated as:

$$P_\theta(\mathbf{z}_n|\mathbf{y_n}, \mathbf{w}_1^{n-1}) = \mathcal{N}(\boldsymbol{\mu}_{\text{prior}}, \Sigma_{\text{prior}}) \tag{4.9}$$

$$P_\theta(\mathbf{w}_n \mid \mathbf{y}_n, \mathbf{z}_n, \mathbf{w}_1^{n-1}) = \prod_{m=1}^{M_n} P_\theta(w_{n,m} \mid \mathbf{y}_n, \mathbf{z}_n, \mathbf{w}_1^{n-1}, w_{n,1}^{n,m-1}) \tag{4.10}$$

When $\mathbf{y}_n$ can be acquired directly, it is intuitive to train a reasonable classifier first and then infer the label from the conversation context. The choices for such classifiers are not limited to simple feed-forward or deep neural networks.

Likewise, the posterior distribution of $\mathbf{z}_n$ can be approximated using the label information as followed by Equation 4.11. Technically,

$$Q_\phi(\mathbf{z}_n|\mathbf{y}_n, \mathbf{w}_1^n) = \mathcal{N}(\boldsymbol{\mu}_{\text{posterior}}, \Sigma_{\text{posterior}}) \tag{4.11}$$

The graphical model is depicted in Figure 4.3. As shown, solid lines denote generative model $P_\theta(\mathbf{z}_n|\mathbf{y}_n, \mathbf{w}_1^{n-1})$ and $P_\theta(\mathbf{w}_n \mid \mathbf{y}_n, \mathbf{z}_n, \mathbf{w}_1^{n-1})$. When y^{t+1} is known, there exists an additional link from y^{t+1} to z (dashed line). C^t encodes context information up to time t. Dotted lines are posterior approximation $Q_\phi(\mathbf{z}_n|\mathbf{y}_n, \mathbf{w}_1^n)$.

The training objective is derived as in Eq. 4.12, which is a lower bound of the logarithm of the sequence probability. When the label is to be predicted ($\bar{\mathbf{y}}_n$), an ad-

ditional classification loss (first term) is added such that the distribution $q_\phi(\mathbf{y}_n|\mathbf{w}_1^{n-1})$ can be learned together with other parameters.

$$\begin{aligned}\log P_\theta(\mathbf{w}_1,\ldots,\mathbf{w}_N) \geqslant\ & \mathbb{E}_{p(\mathbf{w}_n,\mathbf{y}_n)}\left[q_\phi(\mathbf{y}_n|\mathbf{w}_1^{n-1})\right] \\ & - \sum_{n=1}^{N} \mathrm{KL}\left[Q_\psi(\mathbf{z}_n \mid \mathbf{w}_1^n, \mathbf{y}_n)||P_\theta(\mathbf{z}_n \mid \mathbf{w}_1^{n-1}, \bar{\mathbf{y}}_n)\right] \\ & + \mathbb{E}_{Q_\psi(\mathbf{z}_n|\mathbf{w}_1^n,\mathbf{y}_n)}[\log P_\theta(\mathbf{w}_n \mid \mathbf{z}_n, \mathbf{w}_1^{n-1}, \mathbf{y}_n)]\end{aligned} \tag{4.12}$$

4.3.4 Emotion-controlled Response Generation

The major focus in this chapter is to avoid producing generic responses by incorporating emotion information. Hence, we let the label $\mathbf{y}$ indicate the emotion class. When the emotion class can be easily acquired and assigned to the models, no prediction is needed, and thus the training cost does not contain the first item in Formula 4.12. This is the simplest scenario of our framework, which allows explicit control on which class of responses to generate by assigning corresponding values to the label.

As discussed in the introduction, however, it is more realistic that chatbots infer the emotion to express currently. More specifically, the label $\mathbf{y}$ represents the emotion class, which is unknown at test time and needs to be predicted from the context. To achieve this, the probability $q_\phi(\mathbf{y}_n|\mathbf{w}_1^{n-1})$ is modeled by feedforward neural networks. In this case, our framework successfully learns to predict the proper label and decode responses conforming to this label.

4.4 Experiments

4.4.1 Dataset

In order to examine the effectiveness of our framework SPHRED, we create a human-written multi-turn corpus, which contains conversations focusing on the daily affairs. In the following subsections, we will introduce the dataset **DailyDialog** in detail.

A: I'm worried about something.
B: What's that?
A: Well, I have to drive to school for a meeting this morning, and I'm going to end up getting stuck in rush-hour traffic.
B: That's annoying, but nothing to worry about. *Just breathe deeply when you feel yourself getting upset.*
A: Ok, I'll try that.
B: Is there anything else bothering you?
A: Just one more thing. A school called me this morning to see if I could teach a few classes this weekend and I don't know what to do.
B: Do you have any other plans this weekend?
A: I'm supposed to work on a paper that'd due on Monday.
B: *Try not to take on more than you can handle.*
A: You're right. I probably should just work on my paper. Thanks!

Figure 4.4: An Example in DailyDialog Dataset.

In our daily life, there are two main reasons why we communicate with others: *exchange information* and *enhancing social bonding*. In order to communicate and share ideas, we often follow certain dialogue procedures to chat with others. Usually, we don't answer other people's questions rigidly and wait for the next question. Instead, humans usually respond to the previous context first, and then put forward their viewpoints through suggestions or questioning something. By this means, people will focus on what others have said in order to encourage a rapport. Another reason why people chat is to bond up their social connection with others. Hence, daily chats are involve with affections and emotions. By expressing the emotions like happiness and anger, people can show mutual respect, sympathy and understanding with each other, thereby strengthening the connections between each other [148].

We use an example to show the two phenomena mentioned above as in Figure 4.4. We make the *words into italic* to highlight the new information raised by speaker B, which is totally fresh to the other speaker A. Also, we underline and color the words into purple in order to emphasize the expressed emotions. After hearing

from speaker A, speaker B begins worried and shares his/her emotion towards A in the fourth turn of conversation. In order to pacify A, speaker B then proposes to breathe deeply and get away from the upset mood. Such a proposal is new to the conversation context, but it is also dependent and related to the conversation context. This case clearly demonstrates that B's words establish a connection between the preceding history and the ongoing conversations. Note that in the example, some text is omitted due to space limit.

Dataset Construction and Basic Statistics

In order to build a multi-round dialogue dataset, we grab raw data from various websites for English learners to practice oral English dialogues in their daily lives. This is why we call it the **DailyDialog** dataset. The dialogues in the dataset retain the following three attractive characteristics. First of all, the language in DailyDialog is manually written, so it is more formal than the Twitter Dialog Corpus [184] and the Chinese Weibo dataset [240]. The latter consists of posts and replies on social networks, which are noisy, short and different from real conversations. Second, the dialogues in DailyDialog usually focus on a certain topic and a specific physical environment. For example, the conversation that occurs in a store is usually between a customer looking for a suitable product and a salesperson willing to help with the purchase. Another typical conversation occurs between two students talking about summer travel. The third ideal characteristic is that our conversations usually end after a reasonable speaker turns. This distinguishes DailyDialog from existing dialog datasets, such as Switchboard [58] and OpenSubtitles [88], the latter usually has 150+ and 1,000+ speaker turns in one, long-lasting conversation. By studying some examples, we find that in such conversations, people often talk about three or more topics (or scenes). In comparison, our dataset has about 8 speaker turns on average, which is more suitable for training compact conversational models.

After crawling, we de-duplicate the original data, filter out conversations involving more than two participants (three or more speakers), and use the auto-correction package to automatically correct spelling errors.[2] As a result, there are 13,118 multi-turn conversations on the daily topics in our dataset. We also sum up the average token numbers and turn numbers to give a picture of the dataset. The result statistics are present in Table 4.1.

For evaluation, we divide the dataset randomly into a training/validation/test sets, which has 11,118/1,000/1,000 conversations, respectively. The models are trained on the traing set, and the parameters are tuned on the validation set, while the performances are evaluated on the test set.

Table 4.1: Statistics of corpus DailyDialog.

Total Number of Conversations	13,118
Average Speaker Turns Per Conversation	7.9
Average Number Tokens Per Conversation	114.7
Average Number of Tokens Per Utterance	14.6

Annotation Criteria and Procedure

Originally, the dialogues in DailyDialog are crawled from online website for English language learners to master the basic communication ability in daily life. Hence, these dialogues are by nature resembling of the phenomena in daily communications. As discussed in early parts, there are two communication purposes, i.e., *exchanging information* and *enhancing social bonding*. To make full use of the dataset and explore deeply of human conversation behaviors, we carefully annotate the conversations with the perspectives of these two purposes.

The purpose of exchanging information is relevant to the intents, which have been defined as dialogue act or speech at in the previous literature. Generally, speech

[2] `https://github.com/phatpiglet/autocorrect/`

acts stand for the communication functions and objectives when humans converse. Following [4], we annotate the speech acts for each utterance in our dataset using four categories: {Inform, Questions, Directives, Commissive}. The *Inform* category is annotated when the speaker is sharing information and notifying something in the utterance. The *Questions* category indicates that the speaker is seeking and eager to learn information. The *Directives* and *Commissive* categories are counterpart to each other, which includes speech acts like suggest/accept offer. Please refer to [4] for more details on the speech act definition. Afterwards, there are four intent categories defined and annotated in DailyDialog.

The second purpose, enhancing social bonding, is highly correlated with human emotions. The commonly adopted emotion theory is "BigSix Theory" [46], and is also chosen for our dataset. In this case, we annotate each utterance with one of the following emotion category: {Anger, Disgust, Fear, Happiness, Sadness, Surprise}. Besides, we include an extra category {Other} to allow the utterance being annotated with other nuanced categories. As such, there are seven emotion categories in our dataset.

For the sake of labelling quality, we train three experts with professional linguistic knowledge, and ask them to firstly label 100 test samples, then reduce the discrepancy after discussing the criteria details. Finally, these three annotators are required to label the whole dataset independently and reach the inter annotator agreement of 78.9%.

4.4.2 Experimental Setup

Compared Models

In order to examine whether SPHRED is effective on emotion-aware response generation, we adopt two lines of approaches to be compared, i.e., retrieval-based methods and generation-based models. In specific, the compared models are as follows:

- RETRIEVE-EMBEDDING [130]: It is a retrieval method based on similarity scores calculated from embedding space. We measure the distance between embeddings as the average of cosine similarity, Jaccard distance and Euclidean distance. At test time, candidates whose context embedding is closer to the test context embedding are ranked higher. Similar approaches have been adopted extensively on response retrieval task.

- RETRIEVE-FEATURE [84]: It is a retrieval-based method where similarity scores are calculated based on features. We adopt several linguistic features: TF-IDF and three fuzzy string matching features, i.e., QRatio, WRatio, and Partial ratio. We first use TF-IDF to select 1,000 candidates and rank them with the fuzzy features. These fuzzy features are implemented with fuzzywuzzy package.[3] We denote this feature engineering approach as {Feature}. Similar approaches have been demonstrated effectively on response retrieval task and duplicate question detection task,[4] such as [281].

- RETRIEVE-RERANK [130]: It is a two-stage method to encourage the retrieved response to follow a certain criteria. In the first stage, a set of response candidates is retrieved using features, and then fed to the reranker in the second stage. Based on the emotion class, the candidates are reranked and the one with the highest rank is taken as the model output. Specifically, we compare the emotion history of the test example with that of the candidate example, and use the compared similarity as reranking feature. For example, if the test emotion history is {happy,sad,other}, then the candidate response whose emotion history is also (or similar to) {happy,sad,other} will be reranked higher.

- ENC2DEC [218, 190]: The basic cells in this model are all standard GRUs.

[3] `https://github.com/seatgeek/fuzzywuzzy`

[4] `https://github.com/abhishekkrthakur/is_that_a_duplicate_quora_question`

Not that this bare-bones models is excluded with any history utterance, which aims at demonstrating to what extend the performance will be achieved by a standard Seq2Seq conversational agents without knowledge.

- ENC2DEC-ATTN [8]: It is a standard encoder-decoder approach with the widely-adopted attention mechanism. The encoder and decoder in this models are set as GRUs [33] for fair comparison. Note that neither history utterances, nor extra knowledge is incorporated. This model plays a role of benchmarking the performance of Seq2Seq conversational models without knowledge.

- HRED [208]: This state-of-the-art model incorporates history utterances, where a conversation-level ContextRNN is on the top the word-level utteranceRNN.

- VHRED [189]: This model enhances the capability of hierarchical conversational models by including a latent variable to allow more variations in response generation.

We re-implement the aforementioned compared models using TensorFlow [1]. For fair comparison, the vocabulary sizes in all the experiments are set as 25,000, where the out-of-vocabulary (OOV) words are replaced with to a special token UNK. The word embedding size is set as 300. At first, word embeddings are initialized with the Google Word2Vec embeddings.[5] All low-level word-level encoders are defined as 1-layer GRUs with 512 hidden neurons. The high-level context-level encoders in HRED [208] and VHRED [189] are both 1-layer bidirectional GRUs with 1,024 hidden units. We set the minibatch size to 128, and set the learning rate as a fixed number of 0.0002. Adam optimizer [92] is adopted for model training.

[5] `ttps://code.google.com/archive/p/word2vec/`

Evaluation Metrics

In order to systematically compare the model performances, we use two widely-adopted automatic metrics to assess responses produced by the models:

- **BLEU-n**: The N-gram based BLEU scores are proposed to indicate the overlapping degree between the generated responses and the ground-truth response [162];
- **Dist-n**: Since the distinct grams produced by the models stand for the informativeness of the responses, it is reasonable to devise a measurement based on it to evaluate response quality. Typically, the Distinct-n scores stand for the ratios for N-grams [101]. This metric has been widely used in works on response generation [268, 265];

According to the research [119, 156], N-gram based scores like BLEU usually are often inconsistent with human judgments when assessing dialogue models. In order to complement the evaluation, we randomly pick up 100 test samples and perform manual assessments on them. We train three annotators with linguistic background and send the samples to them. To be fair, these annotators have no idea about which model the test response belongs to. The annotators are asked to consider the following 4 aspects when rating the generated response [106]:

- **Relevance**: This measures the relevance degree between the generated response and the input utterances.
- **Fluency**: It assesses the grammar correctness and fluency of the generated responses.
- **Diversity**: This judges the informativeness of the generated responses. If the generated responses contain several repetitive pieces, they are considered as of low diversity.

- Emotion **Appropriateness**: This metric focuses on the emotion expressed in the responses, and evaluate whether they are appropriate to the conversation context.

The rating for each aspect is 3-scale, i.e., {1,2,3}. In specific, a response annotated with 3 for the Fluency aspect means that it is perfectly clear and almost natural; 2 stands for artificial but understandable; 1 is the worst, which means that the generated response is of nonsense. Obviously, the higher of the rating, the better is the generated response.

4.4.3 Performance Evaluation

The experimental results are presented in Table 4.2. We first investigate the performance of retrieval-based methods and the necessity of emotional integration. Since the real response in the test set is not seen in the training set, we cannot use ranking indicators such as Recall-k to evaluate performance. We also do not manually evaluate the retrieved responses because they are all manually written. Instead, we report BLEU-n and Distinct-1 scores obtained by retrieval-based methods. Based on the score, the retrieval-based model in the first block (first three rows) usually produces better results than the generation-based model (the last five rows). Among them, Retrieve-Feature performs the best, which shows that linguistic features are good at capturing semantic similarities. Moreover, Retrieve-Rerank achieves the higher BLEU-2 and BlEU-3 scores than Retrieve-Embedding in the evaluation. Since Retrieve-Rerank model contains emotion information in the re-ranking stage, the experimental results partially verify the benefit of emotion information in conversation modeling.

However, the BLEU-n scores are calculated based on the overlapping degree on the word-level, which do not fully guarantee that the response is suitable for the context of the conversation. Therefore, we also evaluate the performance of retrieval-

Table 4.2: Experimental Results.

	BLEU-1	BLEU-2	BLEU-3	Dist-1	Rel.	Flu.	Divers.	Appr.
Retrieve-Embedding	0.463	0.207	0.162	0.186	-	-	-	-
Retrieve-Feature	**0.477**	**0.258**	**0.204**	**0.205**	-	-	-	-
Retrieve-Rerank	0.426	0.212	0.201	0.186	-	-	-	-
Enc2Dec	0.378	0.156	0.017	0.042	1.46	1.32	1.37	1.65
Enc2Dec-Attn	0.435	0.208	0.017	0.038	1.43	1.38	1.40	1.68
HRED	0.412	0.176	0.020	0.053	1.73	1.55	1.70	1.92
VHRED	0.418	0.181	0.018	0.058	1.83	1.59	1.76	1.79
SPHRED	0.443	0.198	0.021	0.062	2.11	1.72	1.91	2.06

based methods by calculating the percentage of "Equivalence" between the emotion label of the retrieved response and that of the real answer. The results are reported in Table 4.3. Although minor improvements can be seen when using the emotion information, we believe that it is not a strong evaluation indicator, because it still cannot be concluded that the higher the "Equivalence" percentage, the better the retrieved response is.

Table 4.3: Percentage (%) of Emotion "Equivalence" by Retrieval Approaches.

	Embedding	Feature	Rerank
Emotion	69.2	73.7	**74.6**

Then, we examine the performance of generation-based methods and study their effectiveness in using emotion information. The compared generation-based methods can be divided into two categories. The models in the second block (the fourth row to the sixth row) are generative models without any latent variables, and the third block (the last two rows) consists of generation models with additional latent variable. Generally, the performance of those models without latent variables are worse than the model with latent variables. Among the three models in the second block, HRED has the highest BLEU score because it considers history information. When examining the Distinct-1 and Diversity scores achieved by ENC2DEC, ENC2DEC-ATTN and HRED, it is obvious that these models tend to produce universal and boring responses [101, 150] such as "I don't know", which is generally related to most input utterances. These findings are consistent with previous work and prove the necessity of considering more information when developing conversational models.

By examining the last two rows in Table 4.2, we can see that VHRED and the proposed SPHRED are the best two models. Even more, the scores achieved by SPHRED is the closest to the best scores achieved by RETRIEVE-FEATURE, suggesting the high-quality of the responses generated by SPHRED. To analyze deeper,

we compare SPHRED with VHRED. Although they are similar in the way of capturing history utterances, the proposed SPHRED differs VHRED in that: (1) SPHRED utilizes emotion information as an extra label to guide the variable learning, and the learned variable is more meaningful; (2) SPHRED separates the single-line context RNN into two independent parts, which leads to a better context representation. It is also worth mentioning that the size of the hidden states of the context RNN in SPHRED is only half of that in HRED, but SPHRED still yields better performances with fewer parameters. Hence it is reasonable to apply this context information to our framework. In the next subsection, we will investigate on the effectiveness of SPHRED through case studies and ablated experiments.

4.4.4 Analysis

Case Study

In dialogue response generation, word-level overlap metrics such as BLEU-n scores are inadequate to well evaluate the performane of dialogue models [120]. To provide insights on whether emotion information is beneficial, and how it works in the compared models, we firstly conduct a case study using retrieval methods, and presents the results in Table 4.4.

Table 4.4: Case Study of Retrieve-based Approaches.

Test Context
U1: *No way... You can't keep it.* (1) U2: *Please...it's so cute and tame.* (0) U3: *All right. But you have to...* (0)
Retrieved Response
Ground-truth: **I will. Thank you, Mummy.** Retrieve-Feature: *Is there somewhere you wanted to go eat at?* Retrieve-Rerank: *Now we get along very well. It makes me feel...*

We present an example in Table 4.4 to illustrate how emotion improves the quality

of the retrieved responses. Given that the emotion labels in the test context (U1, U2 & U3) of {1, 0, 0}, which means {Anger, Others, Others}, the most appropriate responses retrieved from the emotion-based reranking approach is that "Now we get along very well. It makes me feel that I'm someone special." The context history for this response is "oh, really? so you just took home a stray cat? // Yes. It was starving and looking for something to eat when I saw it. // Poor cat." whose emotion history is {6, 0, 0}, which means {Surprise, Others, Others}.

Ablated Experiment

In the next part of experiments, we aim to examine the effectiveness of the proposed approach in emotion utilization. Since there are potential ways to incorporate emotion information into neural generative models, we compare the proposed SPHRED with several variants. They are:

- EncDec-Emo: We follow [312] to incorporate the label information during decoding. The emotion label is characterized as an one-hot vector. We denote the label-enhanced approaches as {-Emo}.
- HRED-Emo: This variant is similar with EncDec-Emo where the base model is HRED.
- VHRED-Emo: Based on the standard VHRED, emotion information is treated as an extra vector concatenated into decoder states as HRED-Emo does.
- VHRED-Emo+: In this version, both emotion and stochastic variable are taken into consideration. The only difference between this version and the proposed SPHRED is that the base context representation is a single HRED. In other words, this compared model is a simplification of the proposed SPHRED without consideration of speaker uniqueness.

- SPHRED: The proposed full model comprises both speaker and emotion characteristics of conversations.

We also evaluate the models using both automatic and human metrics, and present the results in Table 4.5.

Table 4.5: Ablation Studies.

	BLEU-1	BLEU-2	BLEU-3	Rel.	Flu.	Divers.	Appr.
EncDec-Emo	0.379	0.156	0.018	1.52	1.45	1.54	1.77
HRED-Emo	0.431	0.193	0.016	1.77	1.59	1.76	1.93
VHRED-Emo	0.396	0.174	0.019	1.83	1.44	1.77	1.72
VHRED-Emo+	0.437	0.192	0.018	1.88	1.62	1.81	2.02
SPHRED	**0.443**	**0.198**	**0.021**	**2.11**	**1.72**	**1.91**	**2.06**

By comparing the scores in Table 4.2 and Table 4.5, we can see that introducing emotion information into generation models brings in improvements over EncDec and HRED, but impacts the performance of VHRED. Apparently, VHRED differs from EncDec and HRED in that it owns an extra latent variable. We thus conjecture that the reason behind the performance drop of VHRED-Emo lies in the combination of latent variable and emotion label.

To further analyze the reason, we compare VHRED, VHRED-Emo and VHRED-Emo+. From Table 4.5 we can see that VHRED-Emo lags from VHRED, while VHRED-Emo+ outperforms the standard VHRED. Among these three models, VHRED only has the latent variable, and VHRED-Emo incorporates the emotion label into decoder states without any control over the latent variable, whereas VHRED-Emo+ regulates the latent variable using the emotion information. It is indicated that, simply concatenating extra information with decoder states will sometimes harm the performance. Without any control, the latent variable is confused during the learning especially when an extra emotion information is injected. On the contrary, the

regulation over the latent variable helps yield the performance boost from VHRED and VHRED-Emo+.

The analysis above gives us another hint that, it is instrumental to guide the latent variable learning in varitional models. The guidance from the emotion label in SPHRED is the key to the success of SPHRED. Another contributing component in SPHRED is the separate context representation for each speaker. The best scores yield by SPHRED in Table 4.5 demonstrates that SPHRED not only well keeps individual features, but also provides a better holistic representation for the response decoder than other compared models. To complementary, we also provide some generated responses in Table 4.6.

Table 4.6: Case Study of Generation-based Approaches.

Test Context
U1: *I have to check out today. I'd like my bill ready by 10 in morning.* U2: *You can be sure of that, sir .*
Generated Response
Ground-truth: **Thank you.** HRED-Emo: *all right, sir.* VHRED-Emo: *here you are.* VHRED-Emo+: *okay, fine.* SPHRED: *how long will it take to get there?*

4.5 Chapter Summary

In this chapter, we propose to inject emotion information into open-domain chatbots and examine its benefit for conversation modeling. To verify the proposal, We devise a conditional variational framework for controlled dialogue response generation namely SPHRED. Our framework is novel in that: (1) It models the dialog states for the two speakers separately; and (2) it utilizes information label, i.e., emotion to guide the variable learning. In order to evaluate the effectiveness of the proposed

model SPHRED, we also curate a novel conversation dataset DailyDialog, which is high-quality, multi-turn and manually labeled. The dialogues in the dataset cover totally ten topics and it is rich in emotion. The evaluation results on DailyDialog are indicative: (1) Emotion information in beneficial for both retrieval-based and generation-based conversation models. (2) When introducing emotion information into varional-based models, it should be cautious to control the variable learning. (3) SPHRED is effective in leveraging emotion information to guide the latent variable for conditional response generation.

The proposed SPHRED is also flexible to be applied in real-world scenarios. We only need to adapt the classifier to detect other information, for example, conversation topic, which we leave for future research. External models can also be used for detecting generic responses or classifying emotion categories. In this work, we focus on the controlling ability of our framework. Future research can also experiment with bringing external knowledge to improve the overall quality of the generated responses. Besides, it is also promising to utilize the topic information in DailyDialog dataset by domain adaptation and transfer learning. The proposed dataset is available on `http://yanran.li/dailydialog`, and has been included in the popular huggingface NLP platform as an benchmark conversation dataset.[6] We hope it is beneficial for future research in this field.

[6] `https://github.com/huggingface/datasets/pull/556`

Part II

Conversation-level Coherence

Chapter 5

Knowledge Incorporation for Conversation-level Coherence

5.1 Introduction

In order to generate meaningful responses, social chatbots need to understand the knowledge related to the conversation. Past work has endowed chatbots the ability of entity reasoning mechanism, that is, the ability to refer to the knowledge base (KB) to mention appropriate entities when generating responses [322, 111, 122]. To ensure response quality, it is reasonable that the generated entities should be relevant in terms of semantics and coherent in terms of conversation flow.

Generally speaking, conversation flow is the effortless progression of ideas and responses in a conversation. A natural exchange of inspiration occurs making for a smooth and comfortable experience. In other words, conversation flow happens when conversation is comfortable, effortless and smooth. It is the way conversations are supposed to work. When previous conversation had concentrated more on the actors of a movie, it would be sudden to mention the movie's writer without a smooth transition. Rather, it would be more coherent if the chatbots continue and elaborate more about the actors, and smoothly transit towards other actor-related things. To this end, conversation flow can be reflected by the logic as the conversation goes.

$\mathbf{u}^1$: *Titanic* (film) is really a tragedy.
$\mathbf{u}^2$: So is *Romeo and Juliet* (film).
$\mathbf{u}^3$: *Total Eclipse* (film) is also a sad story.

Figure 5.1: A Conversation Example With Meta-Path.

In this chapter, we aim to explore methods of improving the conversation-level coherence for knowledge-grounded social chatbots. Specifically, we propose to capture the conversation-level coherence through modeling conversation flow. For knowledge-grounded chatbots, the conversation flow is often greatly reflected by the discussion focus, i.e., the entities mentioned during the conversations. Inspired by [44], we model conversation flow by leveraging the meta-paths formed by the entity mentions. A general meta-path is defined as a sequence of object types representing a relationship with the particular semantics. In our case, an example meta-path of mentioned entities could be *actor*→*film*→*film*. To better illustrate how meta-path is indicative for conversation flow, we present a conversation example on Leonardo DiCaprio in Figure 5.1. Linking the mentions to to the equipped KB enables us to map the mentions into their object types, i.e., *Titanic* $\mapsto$ type *film*. We then follow the original order in the conversation to connect the acquired types, resulting in the meta-path *film* $\rightarrow$ *film* $\rightarrow$ *film*. As a result, this conversation example with 3 speaker turns contain a meta-path of $\{\text{F} \rightarrow \text{F} \rightarrow \text{F}\}$, which are implied in the brackets. By using meta-paths, the relationship between two films can be described as *film* $\rightarrow$ *actor* $\rightarrow$ *film* (FAF) and *film* $\rightarrow$ *director* $\rightarrow$ *film* (FDF), where FAF denotes the movies starring the same actor, and FDF denotes the movies directed by the same director.

Because the meta-path information can be treated as a strong indicator of conversation flow, it is intuitive to generate responses and mention the entities following the meta-path to enhance conversation-level coherence. Therefore, we propose a chatbot MOCHA, which is **M**eta-path augmented Kn**O**wledge-grounded **CHA**tbot.

Equipped with an external knowledge base, MOCHA begins by detecting relevant knowledge related to the conversation contexts. Then, MOCHA uses the collected knowledge to understand the conversation context and generates responses with the help of entity candidates. In specific, the encoder transforms the input utterance(s) into an attribute-aware context vector. And the decoder is enhanced with copying mechanism [232] to determine when to copy an entity, and decide which candidate entity to be generated using the newly introduced meta-path information. In particular, 10 most high-frequent meta-paths are defined according to the conversation data, and are then encoded into vectors for model use. Afterwards, MOCHA firstly compares the context vector with each of the learned meta-path vectors, and then selects the candidate entity(s) that complies with the most similar meta-path. Our main contributions are highlighted as follows:

- We propose to model conversation-level coherence by taking into account conversation flow for chatbots, and leverage meta-path information of entity mentions to model conversation flow.
- We augment the knowledge-aware chatbot with meta-path information, and endow it with the awareness of conversation flow information to better capture the conversation-level coherence.
- On two movie conversation corpus, our MOCHA significantly outperforms the compared models and demonstrates the effectiveness of capturing conversation flow through case studies.
- To the best of our knowledge, our work is the first to explore meta-path information in social chatbots.

This chapter is structured as follows. Section 5.2 introduces related work on conversation-level knowledge and meta-path embedding. Section 5.3 presents the

proposed methods on improving conversation-level response coherence. The experimental results and analysis are given in Section 5.4. At last, we summarize this chapter in Section 5.5.

5.2 Related Work on Meta-path and Coherence Evaluation

5.2.1 Meta-path Embedding

Meta-path describes how two nodes in a graph are connected via different types of paths [217]. The measurements or indicators based on the meta-path information in information networks are corresponding to the traditional features. As such, these meta-path based indicators are also potentially beneficial for mining features. The initial attempt along this research area proposes the idea of leveraging meta-path for similarity search. The core is to define the semantic dependencies among different objects, which form meaningful features to the given task. Following this idea, [217] proposes PathSim and compares this newly introduced measure with traditional ones based on random walks. The results clearly show that the meta-path based measure PathSim often yield better performances for the task of finding similar objects in the networks. Another appealing task in the real-world is detecting and predicting the co-authorship relations. To perform the task by utilizing the meta-path information, one can design the meta-paths in the network that are beneficial for the relationship prediction, and learn the weights assigned to the features such as nodes, edges and paths for better results. This idea is later verified by [215], where the most indicative meta-paths are learned and pointed out. This demonstrates that meta-path information is also interpretable when we want to explain the model performances for relation prediction tasks.

In addition to utilizing meta-path for mining information network, [293, 294] em-

ploy the meta-path information to in recommendation systems, where the nodes in the meta-path are constructed by the entities like users and items, and the edges linking the nodes are represented by the user-item consumption records. Based on such correspondences, mainstream recommendation algorithms can be reformulated in the perspective of meta-paths. The most commonly adopted algorithm, item-based collabororative filtering can be represented by meta-paths where the paths capture the user preferences and consumptions towards the items. The content-based recommendation algorithms [164] are the similar. Likewise, it is also feasible to apply the meta-path structure in the recently emergent social-aware recommendations [220].

To the best of our knowledge, our work is the first to explore meta-path information in social chatbots. In our work, we innovatatively utilize meta-path to capture conversation flow, and develop a meta-path augmented chatbot to explore how meta-path will be beneficial for improving the conversation-level coherence.

5.2.2 Coherence Evaluation

There exists a large body of work regarding different notions of coherence and defining coherence from different pespectives. Early approaches to dialogue coherence modeling are built upon available models for monologue, such as the EntityGrid model [11]. As for dialogue models, there are also different perspectives when assessing coherence, such as dialogue act (DA) label coherence, topic and logic coherence. For example, researchers define transition patterns among DA labels [50] associated with utterances to measure coherence. This model restricts utterance vectors only to entity mentions, and needs gold DA labels as its inputs for training as well as evaluation. However, obtaining DA labels from human annotators is expensive and using dialogue act prediction (DAP) models makes the performance of coherence model dependent on the performance of DAP models.

In brief, most of these approaches require human annotated labels [226], which are

difficult to obtain. Even though some of them did not require such labels, they rely on extra resources, such as commonsense knowledge graph [45, 81]. To this end, instead of directly adopting these evaluation metrics, we borrow and combine the ideas from them. For knowledge-grounded models, we borrow the idea from EntityGrid [11], and consider the entity semantic, entity transition and entity consistency coherence at utterance, conversation and context level. For emotion-aware models, we borrow the idea from DA coherence, and consider both conversation and context coherence with respect to intentions.

5.3 Method

5.3.1 Preliminaries

In two-party human-computer conversational systems, chatbots interact with users by returning proper responses. In particular, generation-based conversation models cast the problem of response generation as a Seq2Seq learning problem.

Formally, conversation models take as input the combination of the current user utterance $\mathbf{u}^T$ and conversation histories $\{\mathbf{u}^1, \cdots, \mathbf{u}^{T-1}\}$, where T is the turn number. Each utterance in the conversation is a sequence of words, a.k.a. $\mathbf{u^t} = \{x_1, \cdots, x_{N_t}\}$. Hence, chatbot is fed with a sequence of words $\mathbf{x} = \{x_1, \cdots, x_{N_x}\}$, and is required to generate a response $\mathbf{y} = \{y_1, \cdots, y_{N_y}\}$, where N_x and N_y are the token numbers. When there are multiple turns of previous utterances, the conversation is called multi-round conversation.

Our task is to generate responses according to the user input utterance as well as history utterances, and especially consider the scenario of multi-round response generation. We propose a novel approach to equip the chatbot with topic-based contextual knowledge by linking to a movie knowledge base (MKB) $\mathcal{K}$. The core is how to effectively utilize $\mathcal{K}$ in chatbots. Our approach firstly extracts from $\mathcal{K}$

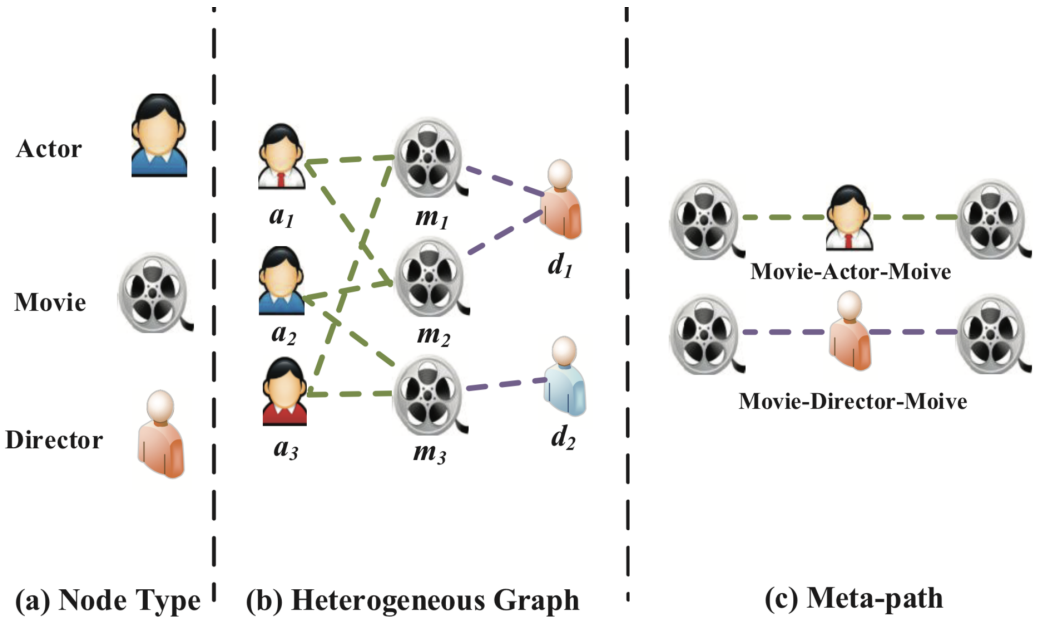

Figure 5.2: Illustration of Meta-path Concepts.

the contextual knowledge specific to the input utterance $\mathbf{x}$, including the discussion attributes and the candidate entities, which are then represented as knowledge embeddings $\mathbf{r}$ and $\mathbf{e}$. The attribute embeddings $\mathbf{r}$ are used when encoding the input $\mathbf{x}$ to produce the conversation context representation, which is then passed to the decoder for response generation. Ideally, a human-like and intelligent chatbot should be aware of multi-round conversation histories, in order to sustain a smooth and natural conversation. The smoothness and naturalness of a conversation can be assessed by the conversation flow. In this chapter, we capture conversation flow by leveraging meta-path information of an equipped KB.

5.3.2 Meta-Path

Formally, we denote the set of entity types as $\mathcal{A}$, and then denote a meta-path $\mathcal{P}$ as $A_1 \rightarrow A_2 \rightarrow \ldots \rightarrow A_{L+1}$, which represents a particular semantic relationship between types A_1 and A_{L+1}, where L is the path length. Giving a meta-path $\mathcal{P}$, there exist multiple specific paths under the meta-path, which is called a path instance denoted

by $\mathcal{P}$.

The illustration of meta-path is shown in Figure 5.2. Take a typical movie knowledge base (KB) for example. There are: (a) nodes with three different types, i.e., actor/actress, movie, and director; (b) a heterogeneous network where the edges with two types are formed by nodes with three types; (c) two meta-paths defined on the KB, i.e., {Movie→Actor→Movie} and {Movie→Director→Movie}.

In addition to pointing out the meta-paths we are interested in, we also need to consider how to quantify the connection between two objects following a given meta-path. Typically, we can use the number of path count, random walk-based measures, or PathSim [217].

Traditionally, the concept of meta-path is introduced to model heterogeneous information networks [216]. A meta-path is defined as a sequence capturing the proximity over its starting and ending nodes. From the perspective of semantic relationships, meta-paths can be interpreted as particular semantics over heterogeneous networks, which permits the researchers to apply meta-path based approaches for real-world network applications.

To the best of our knowledge, our work is the initial attempt to leverage meta-path information to capture conversation coherence for knowledge-aware chatbots. To do so, we observe the conversation data and calculate 10 most high-frequent meta-paths. Each meta-path is formed by 3 connected entity types in their original order during the course of the conversations. In other words, the length of the meta-path we define is 2. Denoting that M (*Movie*), D (*director*), and A (*actor*), the 10 meta-paths we finally regard are present as follows:

- M → M → M
- M → A → A
- M → D → M

- D → M → M
- D → M → D
- D → D → D
- D → A → A
- A → A → A
- A → F → A
- A → M → M

5.3.3 Meta-path Embeddings

After defining these meta-paths, we need to represent the meta-paths in dense forms for model use. Since meta-path is symbolic by nature, the initial step is to generate path instances that are able to capture both the semantic and structural correlations between different types of nodes. To effectively transform the structure of heterogeneous network into skip-gram, however, there is a critical issue that heterogeneous random walks are biased to: (1) high-frequent types of nodes that are prevalent in all counted paths; and (2) centering nodes who are dominant a large amount of paths but are only small-sized in the whole network [217].

To alleviate the issue, [44] proposes the heterogeneous model *metapath2vec*, which injects the structures of heterogeneous network structure into skip-gram, and conducts random walks on the heterogeneous networks based on the newly introduced meta-path information. Similar with the way in skip-gram, metapath2vec regards the nodes with different types as the same, and models distribution of the node frequency without taken into consideration the node types. By this way, the developed meta-path based strategy guarantees the feasibility of incorporating the nodes

with different types into skip-gram model while maintaining the semantic structures among the networks.

Following [44], we generate multiple instances for each path and obtain the instance embeddings using a Gated Recurrent Unit [32]. Because there are a number of valid instances for each 2-length meta-path, the instance embeddings are further pooled into a unified vector. Finally, each meta-path is associated with a continuous vector $\mathbf{p}_m$, where $\forall m \in \{1, \ldots, 10\}$.

5.3.4 Meta-path Augmented Chatbot

After acquiring the meta-path representations in the previous steps, we are now able to augment the knowledge-grounded chatbot with meta-path information, in order to utilize the conversation flow information encoded in the meta-paths. Equipped with an external knowledge base, the augmented chatbot MOCHA consists of three main components:

- An entity collector that retrieves the relevant knowledge from the large KB pool, and prepares the retrieved knowledge for chatbot's later use.
- A context encoder that comprehends conversation context, and represents the context information into dense vectors for decoder's initialization.
- A meta-path augmented decoder that conducts reasoning on the retrieved knowledge and generates the final knowledge-informative response.

Below, we will present each component in detail.

Entity Collector

Given a conversation, MOCHA collects a set of contextual entities E through linking the mentions to the KB. Generally, the knowledge in KBs are stored as triples $\{e_h, r, e_t\}$, where the entities e_h and e_t are connected by the attribute r. Although

there exist numerous facts, the knowledge related to the conversation and necessary to the response generation is limited.

We recognize the entities mentions in a conversation and link them to the KB by [194]. The detected entities are then expanded to form a larger set of entity candidates, in order to facilate larger scope of conversation. Specifically, we follow [111] and regard the detected entities as seeds at the very beginning. Based on the seeds, we collect the neighboring entities within 2-hops. As a result, the candidate set consists of both detected entity mentions as well as their neighbors.

More specifically, we identify the discussion attributes and select candidate entities from $\mathcal{K}$. Four kinds of entities are candidate entities useful for generating responses: the topic film e_τ, the entities explicitly mentioned in the input $e_{\mathbf{x}}$, the entities implicitly mentioned $\hat{e}_{\mathbf{x}}$, and those entities that are new to the input. The last kind of entities, which are denoted as e_r, can be selected based on the detected attributes. In the example, the attribute cast_4 helps select the new entity Spotlight. We recognize $e_{\mathbf{x}}$ by using string matching techniques. However, there often exist multiple mentions for the same entity. To address the coreference problem, we build entity alias dictionaries based on the attribute *alias* to improve matching quality. $\hat{e}_{\mathbf{x}}$ and e_r are selected based on the detected attributes r. As a result, the candidate entities $e = \{e_\tau, e_{\mathbf{x}}, \hat{e}_{\mathbf{x}}, e_r\}$ is formed. We limit the maximum number of it to 10.

The extracted attributes and selected entities are contextual knowledge to supplement the chatbot to respond to the input. However, to use them as discrete features is prone for generalization. To tackle this, we apply TransE [17] to transform $\mathcal{K}$ into knowledge graph embeddings. By retrieving the corresponding embeddings, we can acquire the distributed representations for the extracted attributes and entities as $\mathbf{r}$ and $\mathbf{e}$, respectively. These knowledge embeddings are then fed to the encoder to enhance the input representation and to the decoder to generate engaging responses.

Briefly speaking, the entity collector shortlists a candidate set of contextual entities E. To faciliate the generalization of the model, we apply TransE [17] to encode the discrete-formed candidates into continuous-valued vectors, denoted as $\mathbf{e}_n$, where $\forall n \in \{1, \cdots, N_e\}$. When generating responses, these entity embeddings are used to facilitate entity-aware response generation in the decoder.

Context Encoder

To understand the user input utterance $\mathbf{x}$, we embed utterance tokens using an utterance encoder, and then employ the contextual knowledge collected before to enrich the representation obtained through the encoder.

Typically, we adopt a special variant of RNNs, Gated Recurrent Unit (GRU) [32] as the encoder basis. The GRU cell is formed up by two gates, the update gate $\mathbf{g}_t^z$ and the reset gate $\mathbf{g}_t^r$. In order to consume the information from both directions of the utterance sequences, we use the Bi-directional GRUs, which are indeed two GRUs combined together. One GRU looks forward and the other one looks backward. As a result, each hidden state is concatenated by the representations from both directions, i.e., $\mathbf{h}_t = [\overleftarrow{\mathbf{h}_t}, \overrightarrow{\mathbf{h}_t}]$.

To follow the underlying logic of the discussion, we propose to form the context representation based on the detected attributes. Typically, we use an attribute-based attention mechanism [8] to measure the semantic relevance between the utterance hidden states and the detected attributes. The attribute-attention weights are computed as:

$$\alpha_t \sim \exp(\mathbf{h}_t^T \mathbf{W}_a \bar{\mathbf{r}}) \tag{5.1}$$

$$\bar{\mathbf{r}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{r}_i$$

where $\mathbf{W}_a$ is an intermediate matrix to be learned. Combined with the learned attention, the final context representation $\mathbf{m} = \alpha_t \mathbf{h}_t$, which is then fed to the decoder.

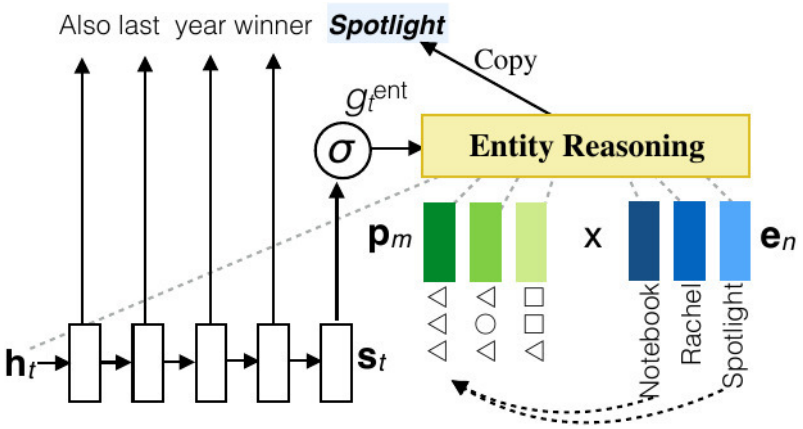

Figure 5.3: Meta-path Augmented Entity-aware Decoder.

Entity-aware Decoder

The last step is to properly respond by using the candidate entities related to the attributes. These candidate entities benefit the response generation when referring is needed.

We augment the decoder with a pointer switch g_t^{ent} to realize the entity-aware generation. The gating variable g_t^{ent} [232] decides whether to generate an entity using p^{ent} or to omit a general word using p^{gru}. The gate g_t^{ent} is trained on:

$$g_t^{\text{ent}} = \sigma(\mathbf{W}_g \mathbf{s}_t)$$

At each time step t, the pointer g_t^{ent} operates like a gate and determines whether to generate a token from the candidate entities or not. When it does, the decoder calculates the probability over the candidate entities to select which one to be mentioned.

At the heart of our MOCHA is how to utilize meta-path information for modeling conversation-level coherence. As shown in Figure 5.3, when the gate is "open", the decoder conducts entity reasoning by taking into account the meta-path information. It firstly approximates how close each meta-path $\mathbf{p}_m$ is to the context $\mathbf{h}_t$, and obtain

the attention weights $\boldsymbol{\alpha}_t$ as:

$$\boldsymbol{\alpha}_t \sim \exp(\mathbf{P}\mathbf{W}_p\mathbf{h}_t)$$

where $\mathbf{P}$ is the matrix consisting of the meta-path vectors $\mathbf{p}_m$. And, $\mathbf{W}_p$ is a learnable matrix that transforms the hidden representations. Then, we apply another attention mechanism on $\mathbf{e}_n$ and obtain the corresponding weights $\boldsymbol{\beta}_t$. Intuitively, the generated entity should belong to the ending type (A_3) in the attended meta-paths. To do so, we align the entity weights with their corresponding path weights, and multiply the two weights as the output probability. Finally, the decoder generates a candidate entity by:

$$p^{\text{ent}}(y_t|\mathbf{h}_t, \mathbf{P}, \mathbf{E}) = \begin{cases} \alpha_{ti}\beta_{tj}, & \text{if } y_t = e_j \text{ and } e_j \mapsto A_i \\ 0, & \text{otherwise} \end{cases}$$

Note that since the meta-path vectors p_m is formed using metapath2vec [44] method by aggregating information from the constitute entity instances, they share the same representation space with the candidate entities e_n. Hence, the multiplication of the two vectors obtained from the above equation indicates the similarity degree of the meta-path and the entity candidate. The higher the similarity, the better the coherence will be achieved after choosing the entity candidate.

When referring is needed, the decoder directly copies the entity with the highest probability. In this way, the generated response is expected to follow the conversation flow by approximating the context representation $\mathbf{h}_t$ with both meta-path information and candidate entities.

5.4 Experiments

5.4.1 Datasets

We examine the proposed approach on two movie conversation corpus. The first corpus we adopt is a publicly available knowledge-driven dialog dataset, DuConv,[1] a carefully-crowdsourced conversation dataset. DuConv [260] is a proactive conversation dataset with 29,858 dialogues and 270,399 utterances. The dialogues are crowd-sourced and formed under a specific requirement. In the two-party conversation, one crowd-sourcer is asked to mainly play the role of a leading player assigned with an explicit goal, a knowledge path comprised of two topics, and is provided with knowledge related to these two topics. The knowledge in this dataset is a format of the triplet {subject, property, object}, which totally contains about 144k entities and 45 properties. We randomly split the dataset by 8:1:1 into training/development/test set.

In addition to this carefully-curated corpus, we also validate the proposed approach on another real-world conversation corpus, Bili-film [111], which is collected from Bilibili, a Reddit-like Chinese movie discussion platform.[2] Although there are other datasets or social platforms, they are either QA-formed or the discussions are too verbose to distill knowledge. Rather, the conversations on BiliBili are more suitable.

To provide external knowledge, we build a movie KB $\mathcal{K}$ based on zhishi.me [155], a Chinese knowledge base with the largest knowledge coverage in movie domain. There are five types of entities in $\mathcal{K}$, i.e., film, director, actor(actress), writer, and genre. We randomly split the corpus by 8:1:1. Finally, 10,000 conversations are used for training, 1,530 for validation, and 1000 for testing. The statistics of DuConv

[1] https://github.com/PaddlePaddle/models/tree/develop/PaddleNLP/Research/ACL2019-DuConv

[2] https://www.bilibili.com/v/cinephile/

Table 5.1: Statistics of Corpus DuConv and Bili-film.

Dataset	DuConv	Bili-film
Total Number of Conversations	29,858	12,530
Total Number of Utterance	270,399	38,467
Average Number of Speaker Turns	9.1	3.6
Average Number of Tokens Per Turn	10.6	27.8
Number of Covered Movies	91,874	187
Number of Covered Movie Stars	51,753	248
Number of Unique Entities Per Conversation	9.3	3.1

and our Bili-film corpus are presented in Table 5.1.

5.4.2 Experimental Setup

For each conversation, we use special symbols "$u" and "$s" respectively for two speakers and place them at the beginning of each utterance. We use Jieba[3] for word segmentation. Following prior work, we construct our KB, collect the candidates and implement the models as described in Chapter 3.

Compared Models

In order to examine whether the proposed approach is effective, we compare our approach with the following state-of-the-art models:

- Attn-Enc-Dec [8]: It is a standard encoder-decoder approach with the widely-adopted attention mechanism. The encoder and decoder in this models are set as GRUs [33] for fair comparison. Note that neither history utterances, nor extra knowledge is incorporated. We choose this bare-bones model to demonstrate to what extend the performance will be achieved by a standard Seq2Seq conversational model without knowledge.

[3] `https://github.com/fxsjy/jieba`

- CONCAT-ENC-DEC [209]: It is a extension of ATTN-ENC-DEC where history utterances are concatenated along with the current input, and still without background knowledge.

- HRED [208]: This state-of-the-art model incorporates history utterances, where a conversation-level ContextRNN is on the top the word-level utteranceRNN.

- FACT-ENC-DEC [54]: This is a knowledge-grounded model that consumes textual "facts" related to the input. To fit it into our scenario, we use the films' one-sentence descriptions as the textual facts. By comparing with it, we aim to distinguish the effects between utilizing unstructured and structured knowledge.

- KB-LSTM [282]: It identifies the knowledge related to the conversation and encodes the knowledge into conversation representation, which is similar with our idea. Differently, KB-LSTM only encodes the entities explicitly mentioned in the input utterance, and incorporates the entity encodings using concatenation operation in the encoder. On the contrary, we feed the context-relevant entities to the decoder for reasoning in response generation while our encoder takes the attribute information into account.

- KB-LSTM+: We improve the above KB-LSTM model by also incorporating the attribute information into the corresponding encoder. This is assumed to inject more knowledge implicitly and thus expand its knowledge scope. We denote this enhanced version as KB-LSTM+.

- GENDS [322]: It is the most similar approach to ours as it also ranks candidate entities collected from the retrieved facts to facilitate entity-aware response generation. The difference is that GENDS lacks explicit mechanism to consider coherence factors.

- MIKE: The proposed model in Chapter 3.

5.4.3 Performance Evaluation

The comparison results of our proposed model and baselines on two datasets are reported in Table 5.2 and Table 5.3. First of all, we can see that the BLEU scores on DUCONV are often lower than those on BILI-FILM. It is due to the linguistic differences of between these two datasets. From Table 5.1, it is obvious that the conversations in DUCONV are longer than those in BILI-FILM, which hinders neural generative models to extract meaningful semantics and obstacles them from producing high-quality responses.

Nevertheless, the model performances are shown similar on these two datasets. The models without extra knowledge (the first block) perform the worst on both datasets, and lag far from knowledge-grounded models (the models in the last two blocks). This finding supports our motivation to incorporate knowledge information into conversation models, which assists chatbots to capture the conversation semantics and in turn form a better reply.

When comparing the models in the second block (the fourth to sixth rows) with those in the third block (the last three rows), we can find that the way to utilize extra knowledge is essentially influential for the chatbot performance. Obviously, FACT-ENC-DEC is the most disappointing one among the models equipped with external knowledge. It is because FACT-ENC-DEC utilizes knowledge described in unstructured text, i.e., *Titanic stars Leonardo as....* Its disappointing performance suggest that it is more effective to inject structural knowledge into Seq2Seq models. Despite of utilizing structural knowledge, KB-LSTM, KB-LSTM+ still generate less satisfactory responses, as indicated by the BLEU-n, Distinct-n and entity-related scores they obtain. While both KB-LSTM+ and KB-LSTM+ employ attribute and entity information, KB-LSTM+ results in negligible improvement (and even decrease) over

Table 5.2: Model Comparison Results on DuConv.

Model	BLEU-2	BLEU-3	Dist-1	Dist-2	Appr.	Gram.	Prec.	Rec.	Coher.
Attn-S2S	0.12	0.08	0.03	0.06	1.64	1.88	0.10	0.07	0.02
Concat-S2S	0.11	0.08	0.02	0.06	1.56	1.81	0.16	0.07	0.04
HRED	0.14	0.09	0.10	0.05	1.68	1.80	0.19	0.10	0.04
Fact-S2S	0.26	0.15	0.13	0.11	1.66	1.83	0.18	0.08	0.05
KB-Lstm	0.37	**0.29**	0.15	0.16	1.72	1.84	0.30	0.17	0.09
KB-Lstm+	0.34	0.23	0.14	0.12	1.68	1.83	0.26	0.15	0.08
GenDS	0.38	0.20	0.14	0.08	1.70	1.79	0.34	0.22	0.17
Mike	0.44	0.29	0.20	**0.18**	1.84	1.82	0.38	0.25	0.26
Mocha	**0.64**	**0.40**	**0.21**	**0.18**	**2.07**	**1.96**	**0.42**	**0.40**	**0.32**

Table 5.3: Model Comparison Results on Bili-film.

Model	BLEU-2	BLEU-3	Dist-1	Dist-2	Appr.	Gram.	Prec.	Rec.	Coher.
Attn-S2S	0.73	0.18	0.03	0.08	1.55	1.79	0.14	0.14	0.03
Concat-S2S	0.76	0.20	0.03	0.10	1.58	1.72	0.14	0.15	0.09
HRED	0.68	0.17	0.02	0.11	1.72	1.34	0.13	0.14	0.10
Fact-S2S	0.82	0.19	0.06	0.16	1.75	1.80	0.13	0.08	0.07
KB-Lstm	1.13	0.32	0.11	0.19	1.82	1.86	0.31	0.23	0.14
KB-Lstm+	1.13	0.37	0.14	0.25	1.82	2.00	0.32	0.31	0.22
GenDS	1.09	0.68	0.16	0.43	1.96	2.03	0.37	0.38	0.25
Mike	**1.27**	0.84	0.19	0.40	**2.40**	**2.15**	0.47	0.53	0.32
Mocha	**1.27**	**0.85**	**0.20**	0.42	**2.40**	**2.15**	**0.50**	**0.58**	**0.36**

the original KB-LSTM.

Remarkably, the three models in the last block, i.e., GENDS, MIKE and the proposed MOCHA win a lot. They three share the same architecture of the decoder and the main difference is how they utilize knowledge in context understanding and entity reasoning. The major findings from the comparison among they three are summarized as follows.

GENDS is the worst among the three models. Different from MIKE, GENDS retrieves entities by matching the fact triples in the KB with the entities explicitly mentioned in the conversation utterances. In such unfiltered way, their candidate set might include noisy entities that are too tangential to the conversation context. As a result, GENDS has larger possibilities of attending on wrong, peripheral entities, and then generates unintelligible responses.

On the contrary, MIKE accesses to new entities E_r linked by the detected attributes. The detected attributes will bias the entity expansion to collect implicit but material entities that closely related to the conversation. This novel strategy enables MIKE to expand the conversation scope, and meanwhile limits the candidate set in a reasonable range.

Our meta-path augmented chatbot MOCHA is consistently better than all the baselines (or achieve the same scores) on the two datasets. The experimental results indicate the effectiveness of MOCHA on open-domain response generation, which adopts a comprehensive way to leverage conversation-level information for improving response coherence. By comparing MOCHA with MIKE, we can find that the overall performance achieved by MOCHA outperforms MIKE, the chatbot developed in the previous chapter. The results show the hierarchical reasoning mechanism is able to successfully utilize the meta-path based context for response generation. Since the importance of each meta-path depends on the context similarity, meta-paths provide beneficial information for the conversation flow among previous utterances, which

has a potential influence on the generated responses.

5.4.4 Analysis

The motivation of this work is to improve conversation-level coherence for response generation. Since conversation flow is formulated as meta-path in this work, we present further analysis to investigate whether conversation flows smoothly by using the proposed method. To do so, we conduct case study and interpret the results vividly.

To give a comprehensive understanding, we sample some generated responses to manually check the model results. In Table 5.4, underlined words are entities copied or referred from the external KB, and input utterances are shorten due to limit space. We translate the original conversation into English for better readability.

There are three sampled cases. In the first conversation, the topic film is *The Notebook*, and the input utterances are focusing on several movies the actress has starred in. After checking the responses, we can see that all the other four models except MOCHA fail to generate informative responses. Even GENDS produces the relevant entity *Rachel*, the entity is the name of the actress. Since both the two speakers are aware of who is the actress they are talking about (as mentioned in the previous utterances), it is weird to repeat the actress name when the focus is the actress's films. Notably, the proposed MOCHA successfully raises a new film starred by the actress, i.e., *La La Land*. This example demonstrates that MOCHA captures a proper conversation flow based on the meta-path information extracted in the conversation context, which is {Film → Film → Film}, a highly frequent meta-path as defined on our dataset.

The second example is more interesting, which is a conversation on the film *Mermaid*. Although the first three models still generate nothing but dull words, we can see that both GENDS and MOCHA produce two entities in one response

Table 5.4: Sampled Generated Responses.

Film	恋恋笔记本	The Notebook
Input	$\mathbf{u}^1$: 女主演过贱女孩, 相当好看 $\mathbf{u}^2$: 推荐你看时空恋旅人，很感动	$\mathbf{u}^1$: The actress also stars in *Mean Girls*, amazing! $\mathbf{u}^2$: I recommend *About Time*, really touching!
Attn	你没看懂	You didn't understand.
HRED	好看好看好看	It is good good good.
Fact	我很喜欢	I like it.
GenDS	是瑞秋	It's Rachel.
Mocha	女主演了很多，还有爱乐之城	The actress stars in a lot of films, such as *La La Land*.
Film	美人鱼	Mermaid
Input	$\mathbf{u}^1$: 女主的演技太让人受不了了 $\mathbf{u}^2$: 只有罗志祥的演技能看	$\mathbf{u}^1$: The actress's acting is intolerable. $\mathbf{u}^2$: Only Zhixiang Luo's acting skill is above average.
Attn	确实很好	Really good.
HRED	女主不好看	The actress is not beautiful.
Fact	电影好	The film is good.
GenDS	周星驰的功夫好	Stephen Chow's *Kung Fu* is good.
Mocha	美人鱼中罗志祥不错	In *Mermaid*, Zhixiang Luo is really good.
Film	敦刻尔克	Dunkirk
Input	$\mathbf{u}^1$: 敦刻尔克是诺兰最好的电影 $\mathbf{u}^2$: 有点夸张，只是诺兰的一次尝试	$\mathbf{u}^1$: *Dunkirk* is the best film of Nolan. $\mathbf{u}^2$: It's overstated. I think this is only a Nolan's experiment.
Attn	我觉得不好看	I don't like it.
HRED	不好	Bad.
Fact	漫威电影	Marvel's film.
GenDS	诺兰黑	I don't like Nolan.
Mocha	敦刻尔克很不错，诺兰的我都喜欢	*Dunkirk* is superb, and I like all Nolan's films.

utterance. In this case, both GenDS and Mocha generate informative responses. When comparing them more carefully, it is obvious that the response generated by GenDS is out of the topic as the entities mentioned by GenDS are less proper to the conversation context. In contrast, the responses uttered by Mocha is more coherent, which echoes the previously mentioned entity *Zhixiang Luo*.

We attribute the cause of the second case to the entity reasoning. At the first glimpse, the entity *Kung Fu* is irrelevant to the topic film *Mermaid*, and *Stephen Chow* does not star in the topic film *Mermaid*. Indeed, *Stephen Chow* is the director of the topic film *Mermaid* who also directs the mentioned film *Kung Fu*. During entity collection step, these two entities will be stored as candidates for response generation. However, GenDS does not has complicated entity reasoning strategy and thus simply select they two to form the final response. As a result, the responses coherence is hampered. In contrast, entity reasoning in Mocha is more considerate. By considering the meta-path information, Mocha filters out the irrelevant entities because their entity types do not correlate with the context. As such, the response generated by Mocha shares better conversation-level coherence. The case in the last example is similar.

5.5 Chapter Summary

In this chapter, we investigate conversation-level coherence using extra knowledge in the structure of meta-path. We propose a meta-path augmented chatbot called Mocha. Built upon the framework Mike proposed in chapter 3, Mocha shares the same knowledge collector and context encoder with Mike. The difference lies in the decoder. Given a conversation, Mocha in this chapter conducts entity reasoning over the pre-collected candidate entities based on the newly introduced meta-path information. Particularly, we define 10 most popular meta-paths observed in our

conversation data, and encode them into vectors using metapath2vec approach [44]. Specifically, when generating responses, MOCHA firstly compares the context representation with each of the meta-path vectors, and then attends on the candidate entities that follow the most similar meta-path. Since meta-path information greatly reflects the conversation flow, the responses generated by MOCHA is expected to be more coherent to the context. On two movie conversation corpus DUCONV and BILI-FILM, we empirically demonstrate the effectiveness of MOCHA.

The major contribution of MOCHA is the incorporation of meta-path information, which takes the conversation flow into consideration and learn effective representations for response generation. Besides the performance improvement, another benefit of the meta-path incorporation is that it makes the generated response highly interpretable. Since meta-paths serve as important interaction context, the attention weights provide explicit evidence to understand why the candidate entity(s) is selected by the model. We further conduct case studies to reveal such interpretability. To the best of our knowledge, our work is the first to explore meta-path information in social chatbots.

Chapter 6

Intention Incorporation for Conversation-level Coherence

6.1 Introduction

Because social chatbots are designed to company users and sustain long, chit-chat conversations, it is critical for them to ensure coherence when generating responses. In other words, the responses should be appropriate according to the conversation contexts, a.k.a. the previous utterances within the current session. The awareness of what have been said will influence the chatbots' behavior on how to respond in the current turn. In conversation modeling, such behavior can be interpreted as a kind of communication intention.

Previous approaches of modeling intention are often designed for task-oriented dialogue systems. Moreover, they rely on specific annotation criteria to categorize the utterances with explicit intention labels, e.g., dialog acts [39]. However, intentions in social conversations are often complicated and implicit, which hinders the feasibility of dialog acts in social chatbots. Alternatively, in this chapter, we consider two primary factors to implicitly capture conversation-level intentions for social chatbots:

- The first factor is *social coherence*. Globally speaking, giving feedback on others' points will generate social coherence [90]. A chatbot enabling of responding

Figure 6.1: A Motivating Example of Social and Individual Coherence.

w.r.t. social coherence will make users feel being well understood and thus fulfill their social needs [200].

- As the conversation goes, the second factor is *individual coherence*. After global considering others, a natural chatbot is also required to perform consistently when defending or elaborating on its own points raised in earlier round(s).

Usually, these two types of intention factors are greatly influenced by the mentioned entities especially in social conversations. As illustrated in Figure 6.1, by responding directly to what has just been mentioned by the user, i.e., *Spotlight*, the chatbot exhibits the sense of social coherence. Individual coherence is revealed in the late of the conversation where the chatbot sticks to its previous ideas about the mentioned entity, i.e., *Spotlight*.

To incorporate these two intention factors, we develop an encoder-decoder architecture enhanced with an entity reasoning mechanism, i.e., the ability to mention proper entities with reference to an associated knowledge base when generating responses. To encourage conversation-level response coherence, we propose to strategically model the aforementioned two factors. On the global level, social coherence is

captured by inter-speaker interactions between two adjacent utterances using a novel interaction unit. Then, individual coherence is handled by keeping two separate entity memories for user and chatbot to ensure speaker consistency. By incorporating these two factors from global to local, our chatbot, namely CHEER, is able to perform **Co**ntext**HE**rence-driven **E**ntity-aware **R**esponse generation. Although coherence has been considered crucial in theoretical conversation analysis [90], previous work on social chatbots captured coherence implicitly by calculating the utterance similarities [318, 141, 297], or based on RL techniques which are hard to train without good supervision [273, 274]. To the best of our knowledge, we are the first to together model social and individual coherence factors, and effectively incorporate them into Seq2Seq conversation models without extra annotation.

We summarize our contributions as follows:

- We identify social coherence and individual coherence as two intention factors in conversation modeling, which have been largely neglected before.
- We propose two carefully designed strategies to model and incorporate these two kinds of coherence into multi-round response generation.
- On two real-world multi-round conversation datasets, we validate the effectiveness of the proposed approach and demonstrate the necessity of intention factors in coherence modeling.

The rest of this chapter is organized as follows. Section 6.2 surveys the previous work on modeling intention for open-domain conversational agents. Section 6.3 describes the proposed method. Experiments and analysis are presented in Section 6.4. Finally, we summarize this chapter in Section 6.5.

6.2 Related Work on Intention-aware Chatbots

Recognizing the importance of context to response coherence, researchers have proposed a wide range of context-aware dialogue models. The easiest way is to use concatenation [127, 209, 113], pooling [209] or weighted combination [223] to integrate history and current utterances as a whole input. A more sophisticated approach is to use a hierarchical encoder by treating the dialogue as a two-level sequence, which has been extended with high-level latent variables to capture the diversity in the dialogue. [264] proposes Sequential Matching Network in which candidate responses are first matched with each pronunciation in the context to accumulate final ranking information. The similarity of [261] is that it uses a separate memory to model each historical utterance, and then uses an additional RNN as a context memory to accumulate its information. However, the reasoning mechanism in [261] is inspired by multi-hop reasoning in reading comprehension tasks and applied to the entire discourse. The difference is that our work suggests that the social coherence between adjacent turns of w.r.t. and the personal coherence of the speaker's turn are mainly integrated into the entity reasoning in the response generation process.

Our work is also related to literature on modeling entities w.r.t. context. [97] proposes to embed entities based on the local contexts of its previous occurrence. Differently, [71] allocates extra blocks of hiddens as memories to track the contexts of entities. While the memories used in their work only models entity implicitly and are updated every timestep, [86] associates with each entity a dynamic representation, which is updated only when an entity appears. Similarly, [284] clusters entity mentions using coreference links and updates the entity states using the last hidden states. Also related is [69] that represents structural knowledge in a dialogue by constructing knowledge graphs whose nodes are updated when being mentioned and being influenced by their neighbors. [313] and [122] combine factual embeddings

with the encoder states, and augment the decoder with copying mechanism as that in [322]. However, none of these approaches considers coherence for multi-round open-domain conversations. Entity consistency has also been demonstrated vital in task-oriented dialogue systems [173, 159]. Recent research points out the significance of generating coherent questions in open-domain dialogue systems [242] and detects identity fraud by asking derived questions [243]. Different from [132] that applies discourse modeling to dialogue coherence, our work develops a global-to-local entity reasoning approach enabling of capturing multi-turn coherence without linguistic annotation. More recently, a few works adopt reinforcement learning to improve dialogue coherence by learning complicated policies [273, 274]. Our work differs with them in at least three folds: (1) We focus on open-domain chatbots rather on a specific information-seeking task; (2) We model multi-turn coherence without linguistic annotation and identify two different coherence factors clearly; (3) Currently our approach is trained with teacher forcing but is potentially compatible with RL techniques, which we leave as future work.

6.3 Method

In this section, we describe the notation and framework of CHEER. The two strategies developed for coherence modeling will be presented in the next section.

Formally, a chatbot generates a response $y = \{y_1, \cdots, y_{N_y}\}$ according to the user input $x = \{x_1, \cdots, x_{N_x}\}$, where N_x and N_y are the token numbers. For multi-round conversation, the chatbot should also consider context information from the history utterance(s) $\{u^1, \cdots, u^T\}$, where $u^t = \{w_{t_1}, \cdots, w_{t_N}\}$, and T is the total turn number excluding the current input.

The proposed CHEER is a knowledge-grounded chatbot equipped with an associate knowledge base (KB) $\mathcal{K}$. Built upon the encoder-decoder architecture, it

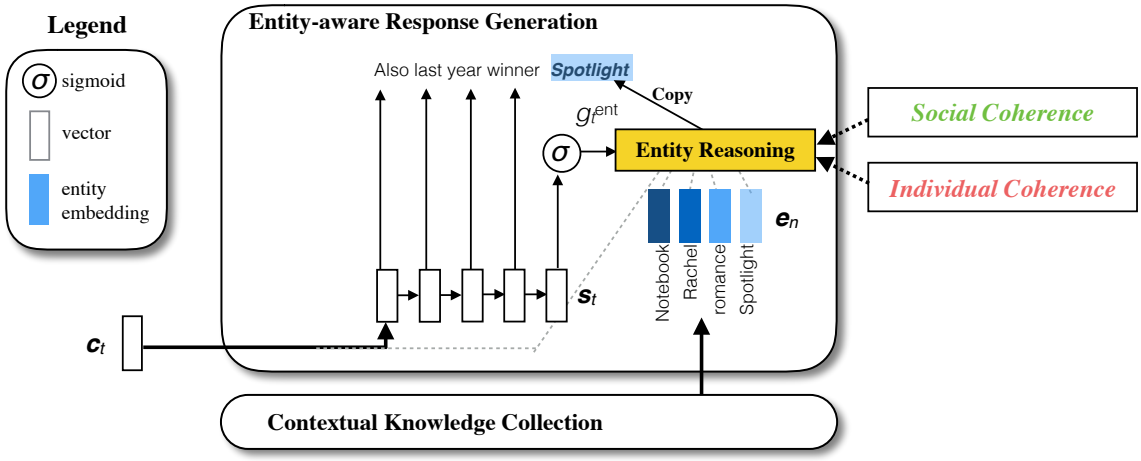

Figure 6.2: Coherence-driven Response Generation via Entity Reasoning.

firstly transforms the input utterance x into context-aware representation, and then conducts entity reasoning by properly referring to the set of pre-collected entities E.[1]

6.3.1 Preliminaries

Specifically, given a conversation equipped with a KB, CHEER models the conversation context using a typical context encoder, which is widely adopted in existing dialogue models. Given an utterance x, we embed its tokens using a special variant of RNNs, bi-directional Gated Recurrent Unit (GRU) [32].[2] In order to consume the information from both directions of the utterance sequences, Bi-directional GRUs are indeed two GRUs combined together. One GRU looks forward and the other one looks backward. As a result, each hidden state is concatenated by the representations from both directions, i.e., $\mathbf{h}_t = [\overleftarrow{\mathbf{h}_t}, \overrightarrow{\mathbf{h}_t}]$.

Based on our preliminary studies, we propose to enrich the representation using the detected attributes to form a knowledge-aware context representation. As shown in the left part of Figure 6.2, we use an attribute-based attention mechanism [8] to

[1]We will explain how to collect the entity candidates in the experiment section.

[2]For multi-round conversation, we concatenate C with x as a single, long utterance. As empirically validated, using hierarchical context encoder did not bring in obvious improvements.

measure the semantic relevance between the utterance hidden states and the detected attributes. We compute the attribute-attention weights as:

$$\alpha_t \sim \exp(\mathbf{h}_t \mathbf{W}_1 \bar{\mathbf{r}}) \tag{6.1}$$

$$\bar{\mathbf{r}} = \frac{1}{N_r} \sum_{m=1}^{N_r} \mathbf{r}_m$$

where $\mathbf{W}_1$ is a learned matrix. Combined with the learned attention, the final context representation $\mathbf{c}_t = \alpha_t \mathbf{h}_t$ is then fed to the response decoder. Intuitively, knowledge-aware context modeling fuses the attribute information into the knowledge-aware context representation, which allows the chatbot follow the underlying logic of the conversation when generating the responses.

The core is to generate proper and informative response according to the user input and the current conversation context. Commonly, response generation is implemented by another GRU that takes as input the context representation $\mathbf{c}_t$ and the previously decoded token y_{t-1} to update its hidden state $\mathbf{s}_t$ [209]:

$$\mathbf{s}_t = \text{GRU}(\mathbf{s}_{t-1}, [\mathbf{c}_t; y_{t-1}]) \tag{6.2}$$

where $[;]$ is the concatenation operator of the two vectors. Then, the decoder uses the hidden state $\mathbf{s}_t$ and the context $\mathbf{c}_t$ to predict the target word y_t at the current time step t through a softmax function:

$$\begin{aligned} p^{\text{gru}}(y_t|y_1, \cdots, y_{t-1}) &= f(y_{t-1}, \mathbf{s}_{t-1}, \mathbf{c}_t) \\ &= \text{softmax}(\mathbf{W}_o \mathbf{s}_t) \end{aligned} \tag{6.3}$$

When referring is needed, response generation can benefit from the set of the collected entities E by directly "copying" the most suitable entity from the set. To achieve this, we augment the decoder with copying mechanism [64]:

$$\begin{aligned} p(y_t|y_1, \cdots, y_{t-1}) &= g_t^{\text{ent}} p^{\text{ent}}(y_t|\mathbf{h}_t, \mathbf{E}) \\ &\quad + (1 - g_t^{\text{ent}}) p^{\text{gru}}(y_t|y_{t-1}, \mathbf{s}_t, \mathbf{h}_t) \end{aligned} \tag{6.4}$$

where $\mathbf{E}$ is the matrix stacking the candidate entity embeddings $\mathbf{e}_n$. The entity gate g_t^{ent} is trained to decide whether to select an entity using p^{ent} or to generate a word from GRU language model using p^{gru}. When the entity gate g_t^{ent} is "open", the chatbot needs to reason which entity(s) is the most proper one(s) to be selected. We regard this problem as **entity reasoning**, and present our solution in the following subsection.

6.3.2 Social Coherence and Individual Coherence

To sustain conversations, the selected entity(s) should not be only relevant, but also coherent regarding the current conversation context. In this part, we elaborate our designs for modeling the two coherence factors, and integrate them into entity reasoning from global-to-local for coherence-driven response generation, which is illustrated briefly in Figure 6.2.

Entity reasoning relies heavily on the understanding of the conversation context. However, the context representation $\mathbf{h}_t$ obtained above is assumed to capture the general word-level semantics of the conversation, which lacks crucial information for response coherence.

To supplement word-level semantics with more information, we propose to explicitly capture the two primary factors: social coherence and individual coherence. *Social coherence* is captured by the inter-speaker interaction between a pair of two adjacent utterances, which puts global effect on how the conversation moves. Formally, we compute the interaction of a pair of two utterances in the successive turns, i.e., $\mathbf{u}^{T-1}$ and $\mathbf{u}^T$ using an interaction function f:

$$\mathbf{d}^T = f(\mathbf{u}^{T-1}, \mathbf{u}^T) \tag{6.5}$$

One plausible way is to re-use the representations obtained by the context encoder described in the previous section, i.e., $\mathbf{u}_t = \mathbf{c}_t$ and compute the word-by-word

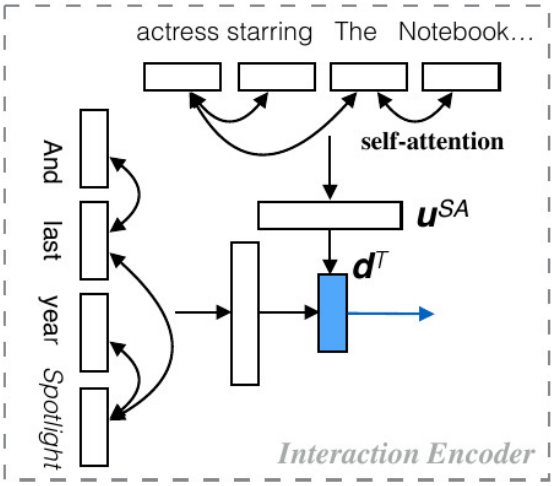

Figure 6.3: Social Coherence By Interaction Encoder.

interaction by alignment:

$$d_{ij}^T = \mathbf{W}_2([\mathbf{u}_i^{T-1}; \mathbf{u}_j^T]) + b \tag{6.6}$$

However, as shown in our experiments, the features learned in this way is not desirable, as they have already been composed by RNNs, resulting in a kind of "leveling effect" on the high-level semantics. In other words, the semantics in $\mathbf{u}_i$ and $\mathbf{u}_j$ carried by RNNs along all previous timesteps are too similar to be used as interaction features. Similar discussions are referred by [228].

Alternatively, we introduce an interaction unit which computes the interaction between the last two consecutive utterances based on their self-attentive representations. As depicted in Figure 6.3, our interaction unit takes as input one utterance represented in the word embedding space, $\mathbf{u}^T = \{\mathbf{w}_1, \cdots, \mathbf{w}_{l_{u^T}}\}$. To capture the semantics from the local context, a self-attention layer on each word pair is computed similar as standard attention [8]:

$$\mathbf{v} = \mathbf{W}_3([\mathbf{w}_i; \mathbf{w}_j; \mathbf{w}_i \circ \mathbf{w}_j]) + b \tag{6.7}$$

$$\boldsymbol{\delta} = \text{softmax}(\max_j \mathbf{v}_{ij}) \tag{6.8}$$

where $\circ$ represents concatenation and element-wise multiplication. The self-attention layer connects each word to any other word in the same utterance. Combing with

a max-pooling operation on the row of $\mathbf{v}$, the self-attention layer outputs a self-attentive representation $\mathbf{u}_{SA}^{T}$, which pays attention to the most indicative words for interaction. We perform similar operations on the other utterance and obtain $\mathbf{u}_{SA}^{T-1}$. Then, the two encoded utterances are fed to the upper-level layers to obtain the interaction features:

$$\mathbf{d}^{T} = f(\mathbf{u}_{SA}^{T-1}, \mathbf{u}_{SA}^{T}) \tag{6.9}$$

where f is the interaction function. Empirically motivated, we adopt a feed-forward neural network. The interaction unit can be regarded as a feature extractor that provides a view for the chatbot to align and cohere with the user.

Upon social coherence, the chatbot is also expected to follow *individual coherence* and maintain self-contained. To achieve this, we explicitly track the entities already been generated by equipping the chatbot with two extra memory units: $\mathbf{k}_{A}^{T}$ stands for the entity just mentioned by the user (A), and $\mathbf{k}_{B}^{T}$ with respect to the latest entity referred by the chatbot (B). Whenever entity reasoning is performed, the corresponding memory will be updated with the latest selected entity embeddings.

6.3.3 Entity Reasoning with Intention Factors

Since response coherence is largely revealed by the mentions of the entities, we incorporate these two factors into entity reasoning. Ideally, the obtained interaction features $\mathbf{d}^{T}$ summarizes global interaction, and the entity memory unit $\mathbf{k}_{B}^{T}$ records the local consistency. Combined with the word-level semantics $\mathbf{c}_t$, the activation of entity reasoning is decided by:

$$g_{t}^{\text{ent}} = \sigma(\mathbf{W}_{s}\mathbf{s}_{t} + \mathbf{W}_{d}\mathbf{d}^{T} + \mathbf{W}_{k}\mathbf{k}_{B}^{T}) \tag{6.10}$$

If the gate is "open", entity reasoning is conducted by attending on the candidate entities using:

$$\boldsymbol{\lambda}^{T} \sim \exp(\mathbf{E}\mathbf{W}_{4}[\mathbf{c}_{t}; \mathbf{d}^{T}]) \tag{6.11}$$

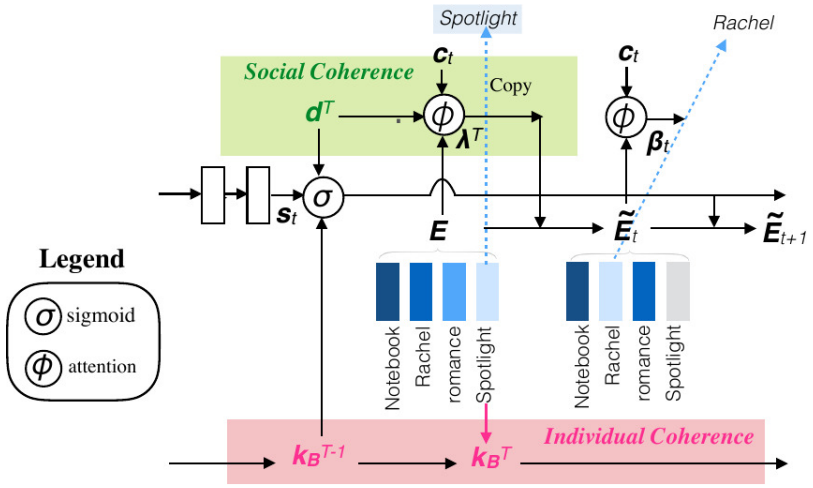

Figure 6.4: Individual Coherence By Memory Units.

After an entity is generated, it is unlikely for it to be used twice in the current response. Inspired by the coverage attention [224], we introduce a vector $\mathbf{q}_t$ to record the probability each candidate entity already been mentioned, where $q_{tj} \in [0, 1]$ that adaptively weights each candidate entity in the embedding space. Initially, $\mathbf{q}_0 = \boldsymbol{\lambda}^T$, which means that the entries corresponding to those attended entities are set to a large value because these entities are probable to be mentioned in the following turns. Intuitively, the value decreases towards 0 when the corresponding entity has been generated. Essentially, the vector $\mathbf{q}_t$ adjusts the candidate entities $\mathbf{E}$ by:

$$\tilde{\mathbf{E}} = ((\mathbf{1}_{N_e} - \mathbf{q}_{t-1}) \otimes \mathbf{1}_k) \circ \mathbf{E} \tag{6.12}$$

where $\mathbf{1}_{N_e} = 1^{N_e}$, $\mathbf{1}_k = 1^k$, and N_e, k is the number and embedding dimension size of the candidate entities, respectively. $\circ$ is the Hadamard product of the two matrices.

The resulted weighted candidate embeddings $\tilde{\mathbf{E}}$ will adapt the attention weights β_{tj} as:

$$\boldsymbol{\beta}_t \sim \exp(\tilde{\mathbf{E}}\mathbf{W}_5[\mathbf{c}; \mathbf{d}^T]) \tag{6.13}$$

To update the weight vector, we utilize g_t^{ent}, the probability of generating a new entity, by:

$$\mathbf{q}_t = \mathbf{q}_{t-1} + g_t^{\text{ent}}\boldsymbol{\beta}_t \tag{6.14}$$

In this way, the decoder has less chance to repeat an entity in the same response. We illustrate the complete mechanism of entity reasoning with two coherence factors in Figure 6.4.

6.3.4 Model Learning

Now the augmented decoder generates a candidate entity by:

$$p^{\text{ent}}(y_t|y<t,\mathbf{E}) = \begin{cases} \lambda_{tj}, & \text{if } y_t = e_j \text{ and } \mathbf{k}_B^T \text{ is empty} \\ \beta_{tj}, & \text{if } y_t = e_j \text{ and } \mathbf{k}_B^T \text{ not empty} \\ 0, & \text{otherwise} \end{cases} \tag{6.15}$$

After the candidate entities are collected already, they serve as the supervision signals to train the switch gate g_t^{ent}:

$$g_t^{\text{ent}} = \begin{cases} 1, & \text{if target word is a candidate entity} \\ 0, & \text{otherwise} \end{cases}$$

To summarize, the gate g_t^{ent} determines when the decoder should copy an entity from the candidates, which is influenced by three factors: the linguistic patterns captured by language model $\mathbf{s}_t$, social coherence reflected by interaction feature $\mathbf{d}^T$, and individual coherence indicated by entity memory $\mathbf{k}^T$. When the gate opens, the decoder attends to the most proper entity based on Eq. 6.15, where the attention weights are globally influenced by $\mathbf{d}^T$ and locally adjusted by $\mathbf{k}^T$.

6.4 Experiments

6.4.1 Datasets

We examine the proposed approach on two movie conversation corpus. The first corpus we adopt is a publicly available knowledge-driven dialog dataset, DuConv,[3] a carefully-crowdsourced conversation dataset. DuConv is a proactive conversation

[3] https://github.com/PaddlePaddle/models/tree/develop/PaddleNLP/Research/ACL2019-DuConv

dataset with 29858 conversations and 270399 utterances. In DuConv, each dialog is formed by two human crowdsourcers, where one human plays the role of leading the conversation, i.e., given related knowledge, initiating a novel topic or continuing the current one in the movie domain [260]. The knowledge in this dataset is a format of the triplet {subject, property, object}, which totally contains about 144k entities and 45 properties. We randomly split the dataset by 8:1:1 into training/development/test set.

In addition to this carefully-curated corpus, we also validate the proposed approach on another real-world conversation corpus, Bili-film [111], which is collected from Bilibili, a Reddit-like Chinese movie discussion platform.[4] Although there are other datasets or social platforms, they are either QA-formed or the discussions are too verbose to distill knowledge. Rather, the conversations on BiliBili are more suitable.

To provide external knowledge, we build a movie KB $\mathcal{K}$ based on zhishi.me [155], a Chinese knowledge base with the largest knowledge coverage in movie domain. There are five types of entities in $\mathcal{K}$, i.e., film, director, actor(actress), writer, and genre. We randomly split the corpus by 8:1:1. Finally, 10,000 conversations are used for training, 1,530 for validation, and 1000 for testing. The statistics of DuConv and our Bili-film corpus are presented in Table 6.1.

6.4.2 Experimental Setup

Preprocessing

For each conversation, we use special symbols "$u" and "$s" respectively for two speakers and place them at the beginning of each utterance. We use Jieba[5] for word segmentation. Following prior work, we construct our KB, collect the candidates and

[4] https://www.bilibili.com/v/cinephile/

[5] https://github.com/fxsjy/jieba

Table 6.1: Statistics of Corpus DuConv and Bili-film.

Dataset	DuConv	Bili-film
Total Number of Conversations	29,858	12,530
Total Number of Utterance	270,399	38,467
Average Number of Speaker Turns	9.1	3.6
Average Number of Tokens Per Turn	10.6	27.8
Number of Covered Movies	91,874	187
Number of Covered Movie Stars	51,753	248
Number of Unique Entities Per Conversation	9.3	3.1

implement the models as described in Chapter 3.

Compared Models

In order to examine whether Cheer is effective in modeling conversation-level coherence, we compare it with several approaches:

- Attn-Enc-Dec [8]: It is a standard encoder-decoder approach with the widely-adopted attention mechanism. The encoder and decoder in this models are set as GRUs [33]. For fair comparison. Note that neither history utterances, nor extra knowledge is incorporated. We choose this bare-bones model to demonstrate to what extend the performance will be achieved by a standard Seq2Seq conversational model without knowledge.

- Concat-Enc-Dec [209]: It is a extension of Attn-Enc-Dec where history utterances are concatenated along with the current input, and still without background knowledge.

- HRED [208]: This state-of-the-art model incorporates history utterances, where a conversation-level ContextRNN is on the top the word-level utteranceRNN.

- T-A-RNN [141]: To improve coherence, this RNN-based model with a dynamic attention model favors the generation of words sharing associations

with salient words in the conversation history. In addition, it incorporates LDA-based topic information following [128].

- FACT-ENC-DEC [54]: This is a knowledge-grounded model that consumes textual "facts" related to the input. To fit it into our scenario, we use the films' one-sentence descriptions as the textual facts. By comparing with it, we aim to distinguish the effects between utilizing unstructured and structured knowledge.

- KB-LSTM [282]: It identifies the knowledge related to the conversation and encodes the knowledge into conversation representation, which is similar with our idea. Differently, KB-LSTM only encodes the entities explicitly mentioned in the input utterance, and incorporates the entity encodings using concatenation operation in the encoder. On the contrary, we feed the context-relevant entities to the decoder for reasoning in response generation while our encoder takes the attribute information into account.

- KB-LSTM+: We improve the above KB-LSTM model by also incorporating the attribute information into the corresponding encoder. This is assumed to inject more knowledge implicitly and thus expand its knowledge scope. We denote this enhanced version as KB-LSTM+.

- GENDS [322]: It is the most similar approach to ours as it also ranks candidate entities collected from the retrieved facts to facilitate entity-aware response generation. The difference is that GENDS lacks explicit mechanism to consider coherence factors.

- CCM [313]: It is a state-of-the-art knowledge graph based conversation model. We use the implementation provided by the authors[6] and fit our KB to their

[6] https://github.com/tuxchow/ccm

setting.

Evaluation Metrics

We evaluate 8 models with four commonly adopted metrics. They are:

- **BLEU-n**: The N-gram based BLEU scores are proposed to indicate the overlapping degree between the generated responses and the ground-truth response [162];
- **Dist-n**: The Dist-1 and Dist-2 scores [101] have been widely used in works on response generation [268, 265];
- 3-scale human evaluation in terms of appropriateness (**Appr.**) and grammatical correctness (**Gram.**) [195];
- precisions (**Prec.**), recalls (**Rec.**) and coherence (**Coher.**) of entities in generated responses to examine the overlapping on referred entities [322] as well as the coherence of the generated entities w.r.t. what has been said in previous rounds. The three metrics (Appr. Gram. and Coher.) are calculated based on 100 manually annotated cases and are used to examine response quality with coherence taken into account. In this case, generating responses that contain irrelevant or inconsistent entities are not preferred.

6.4.3 Performance Evaluation

The experimental results on corpus DuConv and Bili-film are given in Table 6.2 and Table 6.3, respectively. As can be seen, the model performances on the two corpus DuConv and Bili-film are similar. We first examine the importance of entity reasoning in response generation. The worst performances are achieved by the three models in the first block, as indicated by the first four rows on both Table 6.2 and Table 6.3. It is not surprising because they are the models that have no access

to contextual knowledge. Even though T-A-RNN considers multi-round coherence based on a dynamic attention mechanism and LDA-based topic information, its improvement is negligible.

According to whether or not having the mechanism of entity reasoning, the rest five models can be further categorized into two groups (the second and third blocks). Obviously, Fact-Enc-Dec performs the worst, which implies that the unstructured "fact" knowledge representation may impede encoder-decoder models to exploit useful information. The performance of KB-Lstm+ is also unsatisfactory, even though it does incorporate the structural knowledge. KB-Lstm+ comprises the attribute and entity information into a single vector and passes it to the RNN hidden state, which might be too elusive to guide high-quality response generation without entity reasoning. The comparison results on both DuConv and Bili-film demonstrate the superiority of Cheer on incorporating this knowledge.

When examining the performance of the third group of models (the third block), we can see that GenDS, CCM and Cheer outperform the seven models above them on par. The improvement is largely brought by their entity-aware decoders, which allows an explicit mechanism to generate entities from selected candidates. The difference among them is how they understand the context for knowledge reasoning. GenDS [322] retrieves a set of related facts and only uses them to expand the candidate set. CCM [313] includes a knowledge interpreter module, which combines knowledge vectors with utterance embeddings before feeding to the encoder. Considering its unsatisfactory performance, we conjecture that the shallow concatenation in CCM impedes its dynamic graph attention in the decoder to take into account of coherence factors. Overall speaking, Cheer is the best model especially when considering its impressive performance on Bili-film. We attribute the satisfactory performances achieved by Cheer to the success modeling of both social and individual coherence. In Cheer, these two coherence factors are strategically

Table 6.2: Model Comparison Results on DuConv.

Model	BLEU-2	BLEU-3	Dist-1	Dist-2	Appr.	Gram.	Prec.	Rec.	Coher.
Attn-Enc-Dec	0.12	0.08	0.03	0.06	1.64	1.88	0.20	0.07	0.02
Concat-Enc-Dec	0.11	0.08	0.02	0.06	0.04	1.56	1.81	0.16	0.07
HRED	0.14	0.09	0.10	0.05	1.68	1.80	0.19	0.10	0.04
T-A-RNN	0.15	0.08	0.13	0.09	1.68	1.90	0.18	0.07	0.05
Fact-Enc-Dec	0.26	0.15	0.13	0.11	1.66	1.83	0.18	0.08	0.05
KB-Lstm	0.37	**0.29**	0.15	0.16	1.72	**1.84**	0.30	0.17	0.09
KB-Lstm+	0.34	0.23	0.14	0.12	1.68	1.83	0.26	0.15	0.08
GenDS	0.38	0.20	0.14	0.08	1.70	1.79	0.34	0.22	0.17
CCM	**0.47**	0.24	0.12	0.15	1.77	1.81	**0.38**	0.21	**0.24**
Cheer	0.44	**0.29**	**0.20**	**0.18**	**1.84**	1.82	**0.38**	**0.25**	0.23

Table 6.3: Model Comparison Results on Bili-film.

	Automatic Evaluations				Human Judgments				
Model	**BLEU-2**	**BLEU-3**	**Dist-1**	**Dist-2**	**Appr.**	**Gram.**	**Prec.**	**Rec.**	**Coher.**
Attn-Enc-Dec	0.73	0.18	0.03	0.08	1.55	1.79	0.14	0.14	0.03
Concat-Enc-Dec	0.76	0.20	0.03	0.10	1.58	1.72	0.14	0.15	0.09
HRED	0.68	0.17	0.02	0.11	1.72	1.34	0.13	0.14	0.10
T-A-RNN	0.83	0.20	0.04	0.10	1.64	1.75	0.15	0.115	0.09
Fact-Enc-Dec	0.82	0.19	0.06	0.16	1.75	1.80	0.13	0.08	0.07
KB-Lstm	1.13	0.32	0.11	0.19	1.82	1.86	0.31	0.23	0.14
KB-Lstm+	1.13	0.37	0.14	0.25	1.82	2.00	0.32	0.31	0.22
GenDS	1.09	0.68	0.16	**0.43**	1.96	2.03	0.37	0.38	0.25
CCM	1.02	0.66	0.16	0.38	2.01	1.98	0.44	0.38	0.18
Cheer	**1.26**	**0.81**	**0.19**	0.41	**2.40**	**2.15**	**0.53**	**0.56**	**0.34**

incorporated when deciding what to say. Thus, the responses generated by CHEER are more coherent as compared with GENDS and CCM. For better understanding, we show some examples of generated responses for both single- and multi-round conversations in Table 6.4.

6.4.4 Analysis

We conduct additional experiments on corpus BILI-FILM to investigate: (1) how important the two coherence factors are; (2) whether the developed two strategies in CHEER captures coherence effectively. For comparison purpose, we implement a bare-bones model, denoted as CHEER-BASIC.

Social Coherence

To examine the effectiveness of CHEER in modeling social coherence, we compare with:

- GRU-S: As discussed previously, it is plausible to model the inter-speaker interaction by re-using the utterance representation obtained from the knowledge-aware context encoder (See Eq. 6.6). Since the encoder is essentially a GRU, We denote this model as GRU-S.

- CHEER-S: It adopts the proposed interaction unit to capture social coherence. Different from GRU-S, it represents an utterance by applying self-attention mechanism on its composing words.

The comparison results are shown in Table 6.5. For clarity, colored rows are our proposals, and the bold row is the full version. As shown in the first three rows, both GRU-S and CHEER surpasses CHEER-BASIC on the BLEU and coherence scores, indicating the importance of social coherence in response generation. Notably, while GRU-S only achieves negligible improvement, CHEER-S brings in significant

Table 6.4: Sampled Generated Responses.

Single-round Response Generation		
Topic	恋恋笔记本 (The Notebook)	美人鱼 (Mermaid)
Input	男主帅，难怪女主动心了 (The leading actor is so handsome...)	说到画面和特效表现力，这片子水平难道不高吗? (Speaking of art and special effects, isn't it good?)
Attn-Enc-Dec	你没看懂 (You didn't understand.)	别提特效 (Don't mention it.)
Fact-Enc-Dec	男主电影好 (The actor's film is good.)	画面还凑合 (The art is so-so.)
KB-LSTM+	我很喜欢这个电影 (I like this film.)	周星驰的功夫好 (Stephen Chow's *Kung Fu* is good.)
GenDS	是高斯林 (It's Gosling.)	周星驰一张电影票 (Stephen Chow a ticket.)
Cheer	男主演了爱乐之城 (He stars in *La La Land*.)	特效做的西游降魔篇更好 (*Journey to the West* is better.)
Multi-round Response Generation		
Topic	暖暖内含光 (Eternal Sunshine)	大话西游之大圣娶亲 (A Chinese Odyssey Part Two)
Input $\mathbf{x}^1$	男主还演了变相怪杰? (Did he also star in *The Mask*?)	如果我是猴子，我会选白晶晶。 (If I were the monkey, I would choose JingJing.)
Cheer $\mathbf{y}^1$	金凯瑞 (Jim Carrey.)	猴子不是至尊宝，爱紫霞 (Monkey loves ZiXia.)
Input $\mathbf{x}^2$	金凯瑞这么帅的嘛 (So handsome is Jim Carrey!)	还真是两个人，我还是选白晶晶 (I still choose JingJing.)
Cheer $\mathbf{y}^2$	他年轻时很帅 (So he is when he is young.)	我也选晶晶，好悲剧 (I will choose JingJing too, a tragedy.)

Table 6.5: Model Analysis.

Model	BLEU-3	Dist-1	Prec.	Rec.	Coher.
CHEER-BASIC	0.65	0.13	0.41	0.38	0.23
GRU-S	0.69	0.12	0.38	0.38	0.25
CHEER-S	0.76	0.15	0.48	0.49	0.27
SHRED-I	0.62	0.11	0.35	0.39	0.20
CHEER-I	0.67	0.12	0.43	0.50	0.23
Cheer	**0.81**	**0.19**	**0.53**	**0.56**	**0.34**

increases especially on the last three metrics. The performance disparity is resulted from the different representation manners in these two models. GRU-S represents each utterance in a recurrent way, which mixes word semantics together and eludes useful information for interaction modeling. On the contrary, CHEER-S captures social coherence using its self-attentive representation, which highlights salient words in each utterance that are indicative for inter-speaker interaction.

Individual Coherence

To examine the effect of individual coherence, we compare 2 models below:

- SHRED-I [195]: It captures individual coherence using two independent GRU-based encoders to separately model the two speaker states. Its decoder is augmented with copying mechanism for fair comparison.
- CHEER-I: An ablated CHEER where individual coherence is harnessed by two entity memories.

The experimental results are shown in the fourth and fifth rows in Table 6.5. Although SHRED-I performs even worse than CHEER-BASIC, CHEER-I outperforms the baseline model on par. This reveals that individual coherence is beneficial for response generation as long as captured appropriately. In SHRED-I, individual coherence is implicitly encoded in the two separate context GRUs, and the decoder is

expected to distill necessary features from the encoded vectors, which is unreliable. Differently, CHEER-S uses two entity memories to explicitly guide response generation. We thus speculate that it is more effective to handle individual coherence in entity reasoning.

In summary, the overall experimental results demonstrate that the two coherence factors play distinguished and indispensable roles in social conversations. To generate coherent responses, a chatbot needs to take into account coherence both globally and locally.

6.5 Chapter Summary

In this work, we develop a chatbot CHEER to generate coherent responses via entity reasoning by considering both conversation-level intention implicitly. Different from explicit dialog acts, we model two intention factors, social coherence and individual coherence, from global to local. To the best of our knowledge, we are the first to identify and introduce these two intention factors in response generation. On both DUCONV and BILI-FILM corpus, it has been demonstrated crucial to incorporate these two factors, which can be effectively achieved by our two novel designs, i.e., the interaction unit for social coherence and the entity memory for individual coherence.

As discussed before, the intentions in social chatbots are often more implicit as compared with those in task-oriented dialogue systems. It is thus more difficult to define a fixed and reliable annotation criteria to capture the intention in social chatbots. Therefore, in this chapter, we make an initial step to model the intentions based on two communication concepts, i.e., social coherence and individual coherence. Admittedly, the methods developed in this chapter is intuitive but simplified. In the future, we plan to explore more systematized way for modeling coherence in social chatbots.

Part III

Context-level Coherence

Chapter 7

Knowledge Incorporation for Context-level Coherence

7.1 Introduction

To develop non-task-oriented social chatbots [200], existing works based on the Seq2Seq architecture often struggle with the well-recognized "safe response" problem [101] that the generated responses are often too generic to be meaningful.

To mitigate this this, a natural idea is to introduce external information such as knowledge bases (KBs) into response generation. To implement this idea, prior work proposed to devise the response generator with copying mechanism, which allows referring information from KBs [322, 122, 110]. In general, there are thousands of hundreds of possible entities in an equipped KB. It is thus a critical issue to reason which entity(s) is the most proper especially to the whole conversation context.

Essentially, the referred entities should not only be relevant, but also suitable for the input utterances, history conversations as well as the semantics behind the KB. Unfortunately, the approaches used in previous work are not safe to achieve this goal. In specific, the utterance vectors learned using existing approaches [322, 111] only contain information from the conversation side, whereas the entity vectors are acquired using graph embedding models that solely capture the KB network in the

Table 7.1: A Conversation Example.

Turn	Utterances
1	The actress is so amazing in starring also *The Notebook*, *The Vow*.
2	And the last year winner *Spotlight*. Love Rachel so much!
3	Oh my, gorgeous _____!
Candidates:	(1) Rachel McAdams (2) Ryan Gosling

KB side. For utterance representation, it has no idea what is stored in the KB and what is related to a given potential entity. The situation is similar for entity vector since it is unaware of any conversation utterances from the graph embedding models. There exist an information gap between the utterance and entity vectors, which is notorious for context modeling.

We argue that the essential cause is the lack of holistic context understanding. In conversations, people often talk with short and condense expressions, and omit some background knowledge according to the context. It is important but non-trivial to *simultaneously* capture conversation internal information (a.k.a. utterances and histories) and conversation-related external knowledge when we need a comprehensive and precise understanding of the context. In this work, we unify these two kinds of information in one **context graph** (CG). For each conversation, there are two types of nodes in its CG, i.e., utterance and mention nodes.[1] Nodes are also connected with different types of relations. As shown in the example in Table 7.1, when reasoning whether *Rachel McAdams* is a good match, the previous mentions of her starring movies (e.g., *The Notebook* and *The Vow*) are valuable information in the context.

In order to fit the conversation context formed in the unified context graph, we develop a **context graph encoder** suitable for graph understanding, which facilitates a coherent entity reasoning in response generation. Particularly, we identify

[1]In this paper, we will use *mentions* to represent the entities mentioned in conversation utterances, and refer to the entities to be selected as *entity candidates* or *candidates* for short.

the entity mentions in the conversation utterances and link the mentions to the external KB. The extracted entities and conversation utterance themselves serve as the basic nodes in the initial context graph. To process the graph, our encoder begins with calculating the interactions between the entity candidate and each node in the graph, where piece-wise interactions are stored in the node-level vectors. Then, the encoder fuses the node-level features and propagates the features to the graph along the edges, which are aggregated finally to produce an graph-aware candidate vector. Notably, our encoder is built upon the paradigm of the interaction-fusion-aggregation over the stages of node-edge-graph. This allows us to collect the finer features from each node and meanwhile be aware of the useful features from neighboring nodes in the graph. Eventually, a copying-enhanced decoder [322] is adopted to implement the referring mechanism, which approximates the similarity between the CG-aware candidate representation and the conversation.

In brief, our main contributions include:

- We define and model the graph-structured conversation context from history conversations and external knowledge;
- We develop a novel graph-based encoder, namely CGE, that enables holistic conversation understanding;
- We empirically verify the effectiveness of the method as well as each component's contribution.

The remaining of this chapter is organized as follows. In Section 7.2, we conduct survey of related work on modeling conversation context and graph neural networks. Then, we describe the proposed method in Section 7.3. In Section 7.4, we present the evaluation results and experimental analysis. Finally, we summarize this chapter in Section 7.5.

7.2 Related Work on Knowledge Fusion

One essential issue when injecting external knowledge into conversational models is how to fuse the knowledge with conversation text. Traditionally in the task of QA, combination of a KG and a text corpus has been studied with a strategy of late fusion [186, 53] or early fusion [37, 213], which can help address the issue of low coverage to answers in KG based models. With the rapid growth of Graph Convolutional Network (GCN) [95], recent work explores GCNs to efficiently deal with graph-structured data and fuses information into NLP models. A plenty of literature has demonstrated GCN's potential in NLP tasks, e.g., integrating the syntactic and semantic information in neural machine translation [12, 136], text classification [80], question answering [213, 38], word embedding [227] and sequence learning [121]. Please refer to recent reviews [314, 266] for more details. Our work differs from these studies in that we employ a GCN to explore information interactions over the graph-structured conversation context, which is obtained by fusing conversation utterances and an external KB.

Similar to our work is [10]. Notably, Our work is quite different from their model [10] in at least two aspects. First of all, their graph based encoder is only applied to query and dialogue histories, whereas our graph based encoder is developed to understand graph-structured knowledge-enhanced conversation context. Secondly, their model is designated and evaluated for task-oriented dialogue models whereas our model is for open-domain social chatbots. Although GSN [77] and DialogueGCN [55] also leverage dependencies in conversations based on GCNs, they only focus on the information from utterances. In our work, the context graph encoder (CGE) considers heterogeneous interactions among utterances, entity mentions and external knowledge.

7.3 Method

Our aim is to reason whether an entity candidate is suitable to the conversation context.

7.3.1 Context Graph

Given a conversation equipped with a KB, its context graph is formed by two types of nodes: conversation utterances and entity mentions. As demonstrated in Figure 7.1, each blue bubble represents one speaker-turn of utterance, whereas each orange pentagon one entity mention. Green lines with labels depict different types of relations between the nodes. The read star is an entity candidate, and we will explain how to collect the candidates in the experiment section. Specially, the blue bubbles represent utterances $u_j, j \in \{1, \ldots, N_u\}$, whereas orange pentagon stand for entity mentions $m_k, k \in \{1, \ldots, N_m\}$, which is detected using heuristic string matching and entity linking techniques [194]. To keep crucial interactive information, we extract two types of relations among these nodes: *utterance-utterance* relation that links two consecutive utterances from a same speaker, and *utterance-mention* link that connects an utterance node and its consisting mention node. These relations are illustrated as green dash lines in Figure 7.1.

Under the definition, a conversation having N_u speaker-turns of utterances is represented as a context graph $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, with nodes $v \in \mathcal{V}$ and labeled edges (relations) $r \in \mathcal{R}$, where $\mathcal{V} = \mathcal{U} \cup \mathcal{M}$.

7.3.2 Context Understanding using Context Graph Encoder

To understand the graph-structured conversation context $\mathcal{G}$, we propose **Context Graph Encoder** (CGE), based on Graph Convolutional Networks (GCN, [95]). As illustrated in Figure 7.1, each entity candidate c_i (denote as a red star) is fed into

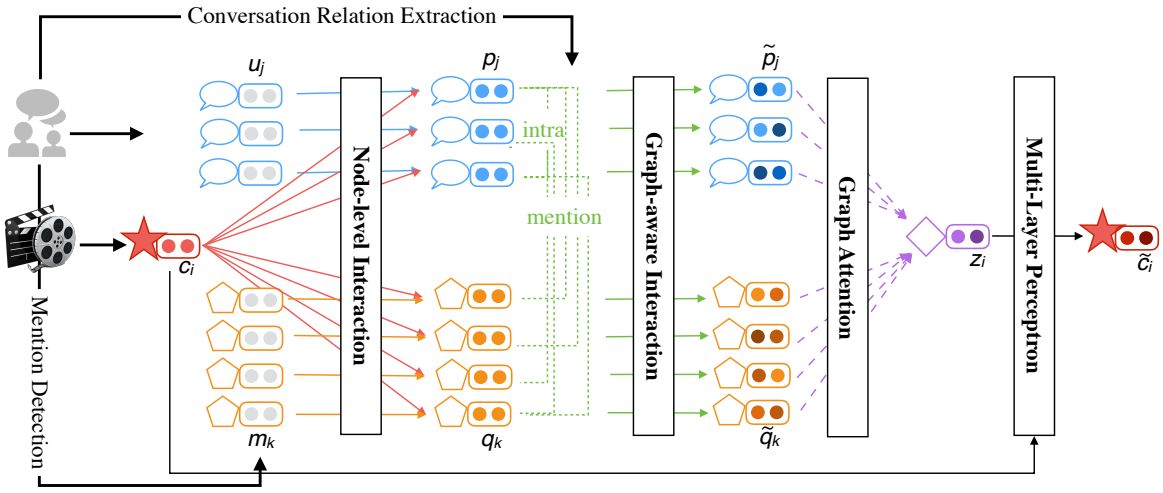

Figure 7.1: Context Graph Encoder.

CGE to fuse graph-structured context information stage-by-stage: (1) Firstly, each node u_j and m_k is computed with candidate c_i to obtain a set of node-level interactions $\mathbf{p}_j$ and $\mathbf{q}_k$; (2) Secondly, a stack of graph convolutional layers is applied to learn graph-aware interactions; (3) These features are then aggregated into a final vector $\mathbf{z}_i$, which captures both local- and global-interactions between the conversation context and the candidate c_i. For each candidate c_i, GCE combines the interaction vector $\mathbf{z}_i$ with its original representation $\mathbf{c}_i$ (i.e., learned from knowledge graph embedding method), resulting in a context-graph-aware representation $\tilde{\mathbf{c}}_i$.

Below, we detail our CGE by starting from initialization followed by the interaction layers.

Node Features

Since the nodes $\mathcal{V} = \mathcal{U} \cup \mathcal{M}$ in $\mathcal{G}$ belong to two types, we initialize them with different manners. Each utterance node u_j in $\mathcal{U}$ is initialized with its consisting word embeddings, $u_j = \{w_1, \ldots, w_{N_j}\} \in \mathbb{R}^K$. To capture the local semantics, a

self-attention layer on each word pair is computed by:

$$\mathbf{u}' = \mathbf{W}_u([\mathbf{w}_a; \mathbf{w}_b; \mathbf{w}_a \circ \mathbf{w}_b]) + b_u \tag{7.1}$$

$$\boldsymbol{\delta} = \text{softmax}(\max_b \mathbf{u}'_{ab}) \tag{7.2}$$

where ; and $\circ$ represents concatenation and element-wise multiplication, respectively. The self-attention layer connects each word to any other word in the same utterance. Combing with a max-pooling operation on the row of $\mathbf{u}'$, the self-attention layer outputs a self-attentive representation $\mathbf{u}_j$, which highlights the most indicative words for interaction. For mention nodes $m_k \in \mathcal{M}$, we initialize them with their corresponding KB embeddings.

Node-level Interaction

.

Note that candidate c_i and nodes u_j, m_k are different objects with distinguished characteristics. Hence, we adopt a bilinear layer to transform them into a shared representation space, and then capture the interactions among them. Without loss of generality, take u_j for example:

$$\mathbf{z}_j = \sigma(\mathbf{W}_{uz}\mathbf{u}_j + \mathbf{W}_{cz}\mathbf{c}_i + b_z) \tag{7.3}$$

$$\mathbf{p}_j = \mathbf{W}_p\mathbf{z}_j + b_p \tag{7.4}$$

where σ is the ReLU activation function [57]. The obtained $\mathbf{p}_j$ are interactive features between utterance node u_j and candidate c_i. By passing the mention node m_k into above equations, we can obtain its corresponding node-level interaction vector $\mathbf{q}_k$.

Graph-aware Interaction

To carry global information in $\mathcal{G}$, we adapt GCN [95] to our problem. In each GCN step, every node in the graph is updated by aggregating its neighboring information along the connecting relations. Taking the step recursively, graph-level

information will be propagated in the graph, and GCN will finally assign a graph-aware feature vector to each node.

For our graph $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, we denote $X \in \mathbb{R}^{|\mathcal{V}| \times Q}$ the node feature matrix, where each row $\mathbf{x}_v \in \mathbb{R}^Q$ is the feature vector for node v ($\mathbf{x}_v \in \{\mathbf{p}_v, \mathbf{q}_v\}$). Then the computation in one GCN layer is defined as:

$$\mathbf{g}_v = f\left(\Sigma_{u \in \mathcal{N}(v)} \left(\mathbf{W}_g \mathbf{x}_u + b_g\right)\right), \forall v \in \mathcal{V} \tag{7.5}$$

where $\mathbf{g}_v$ is the output representation for node v, $\mathcal{N}(v)$ refers to the (immediate) neighbor set of v, f is a non-linear activation function, and we empirically adopt ReLU [57].

Considering the typed-relations among the nodes, we expand the edge set $\mathcal{R}$ by incorporating the inverse-edges and self-loops, and denote the expanded set as $\mathcal{R}'$. Then, we devise the recursive computation of stacking multiple GCN layers as:

$$\mathbf{g}_v^{l+1} = f\left(\Sigma_{u \in \mathcal{N}(v)} \left(\mathbf{W}_{r_{uv}}^l \mathbf{g}_u^l + b_{r_{uv}}^l\right)\right) \tag{7.6}$$

Note that the trainable parameters $\mathbf{W}_{r_{uv}}^l$ and $b_{r_{uv}}^l$ are layer- and typed-relation specific. After applying the devised GCN layers for several hops, we obtain the graph-aware representation of p_j and q_k, and denote them as $\tilde{\mathbf{p}}_j$ and $\tilde{\mathbf{q}}_k$, where $\tilde{\mathbf{p}}_j, \tilde{\mathbf{q}}_k \in \mathbf{g}_v^L$, and L is the number of total GCN layers.

Interaction-aware Representation

The proposed CGE follows the interaction-fusion-aggregation paradigm. To aggregate the information in the context graph, we adopt attention mechanism similar

as [229]:

$$\mathbf{z}_i = \sum_{n=1}^{N_u+N_m} \beta_n \mathbf{g}_n^L \tag{7.7}$$

$$\beta_n = \frac{\exp(\omega_n)}{\sum_{j=1}^{N_u+N_m} \exp(\omega_j)} \tag{7.8}$$

$$\omega_n = \mathbf{h}_a^{\intercal}(\tanh(\mathbf{W}_a \mathbf{c}_i + \mathbf{W}_b \mathbf{g}_n^L)) \tag{7.9}$$

where $\mathbf{h}_a$ is trainable attention vector. With the learned attention weights, we obtain the final interaction vector $\mathbf{z}_i$ considering the importance of each node in the context graph.

Until now, for each candidate c_i, we have two different representations: its original representation $\mathbf{c}_i$ (learned using graph embedding methods), and its interactive representation $\mathbf{z}_i$ (obtained after Eq. 7.9). Since they two provide information from different perspectives, we borrow the idea from [149] and combine the representations via:

$$\tilde{\mathbf{c}}_i = \tanh(\mathbf{W}_f[\mathbf{c}_i; \mathbf{z}_i; \mathbf{c}_i \circ \mathbf{z}_i; \mathbf{c}_i - \mathbf{z}_i] + b_f) \tag{7.10}$$

where $\circ$ denotes the element-wise product.

Intuitively, the fused representation $\tilde{\mathbf{c}}_i$ shares semantics from both external KB and conversation utterances, and highlights the fine-grained interactions among the corresponding context graph (CG) through layers of GCN propagation. In this way, the proposed CGE provides holistic understanding of the conversation context with the help of CG, generates CG-aware representations of knowledge, which will in turn facilitate response generation.

7.3.3 Response Generation

To generate proper and informative response, we adopt GRU [32] as the decoder basis. The GRU cell is fed with the context representation $\mathbf{h}_t$ and the previously

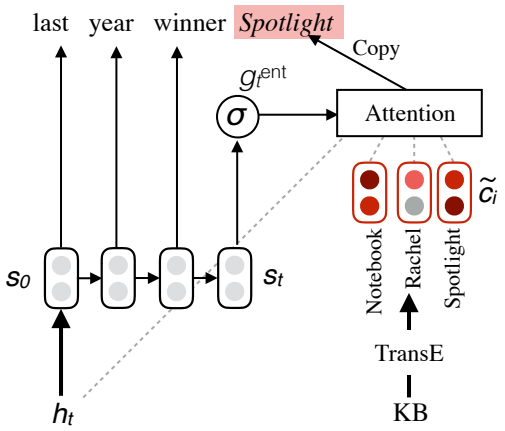

Figure 7.2: Knowledge-grounded Response Decoder.

decoded token y_{t-1} to update its hidden state $\mathbf{s}_t$:

$$\mathbf{s}_t = \text{GRU}(\mathbf{s}_{t-1}, [\mathbf{h}_t, y_{t-1}]) \tag{7.11}$$

$$\mathbf{h}_t = [\overleftarrow{\mathbf{h}_t}, \overrightarrow{\mathbf{h}_t}] \tag{7.12}$$

where $[;]$ is the concatenation operator of the two vectors. We obtain $\mathbf{h}_t$ by recurrently encoding the input utterances using another bi-directional GRU.[2] By recurrently processing the state vectors, the decoder generates the response by conducting a softmax function over the vocabulary:

$$\begin{aligned} p^{\text{gru}}(y_t|y_1, \cdots, y_{t-1}) &= f(y_{t-1}, \mathbf{s}_t, \mathbf{h}_t) \\ &= \text{softmax}(\mathbf{W}_o \mathbf{s}_t) \end{aligned} \tag{7.13}$$

Following previous work [322, 313, 289], we augment the decoder with copying mechanism [64]. In this way, response generation can benefit from a set of pre-collected entity candidates C by directly "copying" the most suitable candidate from the set. In specific:

$$\begin{aligned} p(y_t|y_1, \cdots, y_{t-1}) &= g_t^{\text{ent}} p^{\text{ent}}(y_t|\mathbf{h}_t, \mathbf{C}) \\ &\quad + (1 - g_t^{\text{ent}}) p^{\text{gru}}(y_t|y_{t-1}, \mathbf{s}_t, \mathbf{h}_t) \end{aligned} \tag{7.14}$$

[2]For multi-round conversation, we concatenate the utterances into one, unified utterance. As empirically validated, using hierarchical context encoder did not bring in obvious improvements.

where $\mathbf{C}$ is the matrix stacking the candidate embeddings $\tilde{\mathbf{c}}_i$. Note that we use the context-graph-aware embeddings provided by the proposed CGE. When the entity gate g_t^{ent} is "open", the decoder approximates how close each candidate is to the context using attention mechanism [8]:

$$\boldsymbol{\alpha}_t \sim \exp(\mathbf{C}\mathbf{W}_c\mathbf{h}_t) \tag{7.15}$$

Otherwise, the decoder switches back to a vanilla GRU language model and omits a general word based on the softmax output. The gate g_t^{ent} is trained on the hidden state:

$$g_t^{\text{ent}} = \sigma(\mathbf{W}_e\mathbf{s}_t) \tag{7.16}$$

By utilizing the augmented decoder, knowledge-grounded response generation refers to an KB entity by:

$$p^{\text{ent}}(y_t|y_{t-1}, \mathbf{s}_t, \mathbf{h}_t, \mathbf{C}) = \begin{cases} \alpha_{ti}, & \text{if } y_t = c_i \\ 0, & \text{otherwise} \end{cases} \tag{7.17}$$

Since the candidate embeddings $\tilde{\mathbf{c}}_i$ have been augmented using the proposed CGE, they are aware of comprehensive information over the graph-structured conversation context. These CG-aware candidate embeddings thus allow more accurate matching between the candidate and the conversation (Eq. 7.15). As a result, the decoder will attend to more proper entity(s), which will be demonstrated through our experiments.

7.4 Experiments

7.4.1 Datasets

As before, we examine the proposed approach on two corpora for knowledge-grounded conversation model evaluation, DuConv and Bili-Film.

7.4.2 Experimental Setup

Preprocessing

For each conversation, we use special symbols "$u" and "$s" respectively for two speakers and place them at the beginning of each utterance. We use Jieba[3] for word segmentation. Following prior work, we construct our KB, collect the candidates and implement the models as described in Chapter 3.

Compared Models

We compare with the following 10 models to examine the effectiveness of the proposed graph-structured chatbot:

- ATTN-S2S: The sequence-to-sequence (S2S) approach with a vanilla attention mechanism [8].
- CONCAT-S2S [209]: It is a extension of ATTN-S2S where history utterances are concatenated along with the current input, and still without background knowledge.
- HRED [208]: This state-of-the-art model incorporates history utterances, where a conversation-level ContextRNN is on the top the word-level utteranceRNN.
- HGFU [287]: Hierarchical Gated Fusion Unit (HGFU) incorporates a cue word extracted using pointwise mutual information (PMI) into the decoder to generate meaningful responses.
- FACT-S2S [54]: A knowledge-grounded conversation model that consumes unstructured facts. We use the films' one-sentence descriptions as the textual knowledge.

[3] `https://github.com/fxsjy/jieba`

- KB-LSTM [283]: This model comprises the entities explicitly mentioned in the input utterance into the encoder. To expand its knowledge scope, we enhance it by attributes, denoted as **KB-LSTM+**.

- GENDS [322]: This model shares similar decoder with ours, and ranks entity candidates collected from the retrieved facts.

- CCM [313]: It is a state-of-the-art knowledge graph based conversation model. We use the implementation provided by the authors[4] and fit our KB to their setting.

- CHEER: It is a conversation model proposed by ourselves in order to capture conversation-level coherence through modeling intention factors implicitly.

7.4.3 Performance Evaluation

The analysis of the performances in the first two blocks are similar with Chapter 5. Here we directly move the attention towards the third block, where the model GENDS and CCM yield satisfatory performances when comparing them with the other seven models in the first and second blocks. We deduce such good performance to the incorporation of the copying-allowed decoders GENDS and CCM. This is consistent with the analysis in the previous chapters. Here, we are also curious the reason why CCM is not as good as expected. Even though CCM includes knowledge interpreter module which is able to learn certain interactions among knowledge entities, we suppose that it only learns shallow and vague interactions, which are not enough and even notorious to context understanding when the interactions are wrongly captured.

[4] `https://github.com/tuxchow/ccm`

Table 7.2: Model Comparison Results on DuConv.

Model	BLEU-2	BLEU-3	Dist-1	Dist-2	Appr.	Gram.	Prec.	Rec.	Coher.
Attn-S2S	0.12	0.08	0.03	0.06	1.64	**1.88**	0.10	0.07	0.02
Concat-S2S	0.11	0.08	0.02	0.06	1.56	1.81	0.16	0.07	0.04
HRED	0.14	0.09	0.10	0.05	1.68	1.80	0.19	0.10	0.04
HGFU	0.12	0.09	0.09	0.04	1.43	1.25	0.21	0.08	0.07
Fact-S2S	0.26	0.15	0.13	0.11	1.66	1.83	0.18	0.08	0.05
KB-Lstm	0.37	**0.29**	0.15	0.16	1.72	1.84	0.30	0.17	0.09
KB-Lstm+	0.34	0.23	0.14	0.12	1.68	1.83	0.26	0.15	0.08
GenDS	0.38	0.20	0.14	0.08	1.70	1.79	0.34	0.22	0.17
CCM	0.47	0.24	0.12	0.15	1.77	1.81	0.38	0.21	0.24
Cheer	0.44	0.29	0.20	0.18	1.84	1.82	0.38	0.25	0.23
CGE	**0.51**	0.28	**0.19**	**0.22**	**1.80**	**1.88**	**0.41**	**0.26**	**0.33**

Table 7.3: Model Comparison Results on Bili-film.

Model	BLEU-2	BLEU-3	Dist-1	Dist-2	Appr.	Gram.	Prec.	Rec.	Coher.
Attn-S2S	0.73	0.18	0.03	0.08	1.55	1.79	0.14	0.14	0.03
Concat-S2S	0.76	0.20	0.03	0.10	1.58	1.72	0.14	0.15	0.09
HRED	0.68	0.17	0.02	0.11	1.72	1.34	0.13	0.14	0.10
HGFU	0.64	0.19	0.02	0.10	1.76	1.49	0.19	0.17	0.11
Fact-S2S	0.82	0.19	0.06	0.16	1.75	1.80	0.13	0.08	0.07
KB-Lstm	1.13	0.32	0.11	0.19	1.82	1.86	0.31	0.23	0.14
KB-Lstm+	1.13	0.37	0.14	0.25	1.82	2.00	0.32	0.31	0.22
GenDS	1.09	0.68	0.16	0.43	1.96	2.03	0.37	0.38	0.25
CCM	1.02	0.66	0.16	0.38	2.01	1.98	0.44	0.38	0.18
Cheer	1.26	0.81	0.19	0.41	2.40	**2.15**	0.53	0.56	0.34
CGE	**1.32**	**0.90**	**0.19**	0.42	**2.46**	2.13	**0.52**	**0.60**	**0.43**

Table 7.4: Model Analysis.

Model	BLEU-3	Dist-1	Prec.	Rec.	Coher.
BASE	0.65	0.13	0.41	0.38	0.23
+NI	0.72	0.15	0.44	0.46	0.27
+GI	0.86	0.19	0.50	0.55	0.36
+GA	**0.90**	**0.19**	**0.52**	**0.60**	**0.43**

Overall speaking, CHEER and CGE are the best two models (the last two rows). We attribute the satisfactory performances achieved by CHEER to the consideration of conversation-level coherence. In CHEER, two kinds of intention factors are strategically incorporated when deciding what to say. Thus, the responses generated by CHEER are more coherent as compared with GENDS and CCM. Despite that CHEER models two intention factors explicitly, the knowledge representation and context representation in it are learned separately, which hampers the performance of entity reasoning. On the other side, CCM makes it in learning a comprehensive representation based on the graph structure conversation context. Utilizing the highest scores obtained by CGE, it proves the need to understand the context of the conversation from a holistic perspective when generating responses. In general, the proposed CGE is significantly better than the comparative model for all indicators except Dist-2. In particular, the automatic Distinct-n score and knowledge coverage score (the three rightmost columns) indicate that the responses generated by CGE are more diverse, smooth and consistent with the context of the conversation.

7.4.4 Analysis

Ablation Studies. We conduct extra experiments to answer a question: whether each component in the developed CGE is beneficial for comprehensive context modeling. To do so, we build up a bare-bones model BASE, in which there is no context information used.

Table 7.5: Sampled Generated Responses.

Single-round Response Generation		
Film	恋恋笔记本 (The Notebook)	美人鱼 (Mermaid)
Input	男主帅，难怪女主动心了 (The leading actor is so handsome...)	说到画面和特效表现力，这片子水平难道不高吗? (Speaking of art and special effects, isn't it good?)
Attn-S2S	你没看懂 (You didn't understand.)	别提特效 (Don't mention it.)
Fact-S2S	男主电影好 (The actor's film is good.)	画面还凑合 (The art is so-so.)
KB-Lstm+	我很喜欢这个电影 (I like this film.)	周星驰的功夫好 (Stephen Chow's *Kung Fu* is good.)
GenDS	是高斯林 (It's Ryan Gosling.)	周星驰一张电影票 (Stephen Chow a ticket.)
CGE	男主还演了爱乐之城 (He also stars in *La La Land*.)	比西游降魔篇好看，这次特效更好 (*Journey to the West* is better.)
Multi-round Response Generation		
Film	暖暖内含光 (Eternal Sunshine)	烈日灼心 (The Death End)
Input x^1	男主还演了变相怪杰? (Did he also star in *The Mask*?)	邓超演技这么好，后来却走上了喜剧的路 (Chao Deng's acting is splendid, why he became a comedian?)
CGE y^1	是金凯瑞 (It's Jim Carrey.)	我觉得他在影里表现惊艳 (He performed impressive in *Shadow*.)
Input x^2	金凯瑞这么帅的嘛 (So handsome is Jim Carrey!)	演得太好了，两个人物都厉害 (He acted the two characters both brilliant.)
CGE y^2	金凯瑞年轻太帅了 (Jim Carrey is young and handsome.)	是孙俪一起的 (The film is also acted by Li Sun.)

There are three stages in CGE: (1) Node-level Interaction (NI) refers to Eq. 7.4; (2) Graph-level Interaction (GI) refers to Eq. 7.6; and (3) Graph Attention (GA) refers to Eq. 7.9. Adding them step-by-step on the top of the bare-bones model BASE, we obtain several variants of the proposed CGE. It is clearly revealed in Table 7.4 that all the developed component is contributing to the final performances.

Case Studies. We also present qualitative analysis in addition to quantitative evaluation, in order to study why our approach works and where future work could potentially improve it. Some generated responses are present in Table 7.5, where the underlined words are entities. Note that input utterances are shorten due to limit space.

In the single-round conversation setting (the upper block), take the film *The Notebook* as example. Despite that GENDS generates a related entity Ryan Gosling, it is based on the similarity between utterance representation and the KB embedding of Ryan Gosling. This best match makes the response less informative, since it only reveals the name of the leading actor. On the contrary, the proposed CGE fuses context graph information into the KB candidates, which in this case allows the embedding of Ryan Gosling aware of conversation context: who is the leading actor, and what the conversation talks about. Thus, CGE successfully refers to another film *La La Land* acted by the leading actor. Considering the context, this new entity is beneficial to sustain future conversations.

Similar behavior is also observed in the multi-round conversation setting (the lower block). Take *Eternal Sunshine* as example. When generating y^2, there are three utterance nodes (i.e., x^1,y^1,x^2) and two mention nodes (i.e., The Mask and Jim Carrey) in the context graph. By learning the interactions over the graph and discriminating the importance of the nodes, CGE captures the focus as the conversation goes, and replies appropriately.

Nevertheless, there is still room for improvement. The response y^2 generated

by CGE in the case *Eternal Sunshine* seems repetitive to the conversation context. The potential reason is that entity Jim Carrey has been mentioned several times in the context, which leads CGE pay too much attention on it. In the future, it is promising to explore ways of balancing knowledge novelty and coherence.

7.5 Chapter Summary

Previous work on building knowledge-aware chatbots often target at improving the method of knowledge incorporation and reasoning for response generation. To do so, existing approaches often devise decoders in Seq2Seq chatbots. We argue that knowledge is also and even more important for conversation understanding. However, how to incorporate knowledge into the encoders is less explored. To investigate the significance of knowledge in conversation understanding, in this work, we develop a graph-based encoder, CGE. The proposed encoder is operated on the graph-structured context to fuse the information from both conversation utterances and external knowledge, and finally obtain a comprehensive understanding for the whole conversation context. On two large-scale conversation corpora, it has been demonstrated crucial to reason knowledge by considering conversation utterances and KB together. We also empirically validate the effectiveness of the proposed approach through both quantitative and qualitative evaluations.

In the future, there are two directions we are interested to explore. Currently, pre-training models are promising, and they are expected to learn implicit linguistic knowledge from large-scale datasets. Since our method is compatible with pretrained models in theory, it is promising to combine pre-training methods with the proposed CGE and develop a pretrained knowledge-grounded model. Secondly, the main idea of our method is to fuse information from different sources, and model conversations in a holistic view. Hence, it is also interesting to incorporate both structured and

unstructured knowledge to further enhance our chatbot.

Chapter 8

Emotion and Intention for Context-level Coherence

8.1 Introduction

As research has shown, it will increase the user's participation and satisfaction, thereby giving the dialogue agent emotional intelligence, which is an important human intelligence. Previous works in this field only use emotion information through a gating mechanism [312] and directly integrate the emotion vector into the response decoding, which may not be sufficient.

Indeed, various potential factors that affect people's daily life. When friends express their upset moods, people usually sympathize with them and ask why [139]. Humans are inherited with shared mental states (such as empathy), which contribute greatly to emotional intelligence. In daily life, these mental states stimulate our communication behaviors when we chat with others. It is reasonable to model these factors together to obtain a context-level understanding of the speaker's understanding. In other words, ensuring response consistency at the context level requires comprehensive modeling of many factors. **Emotion** and **intention** are two necessary factors. Take the two dialogues in Fig 8.1 as an example. Person B receives a dance invitation from A, but shows different wishes under certain emotions. In the

A: **Would you like to go dance with me tonight? (happy, question)**
B: *Sounds great! Come by any time!* (happy, inform)
A: Cool. Let's meet at the club on 8 p.m. (happy, directive)
B: No problem, see you then. (happy, commissive)
- -
B: *I am just not in the mood for this.* (sad, inform)
A: You look so upset. What's going on? (others, question)
B: I lost the table tennis game yesterday. (sad, inform)

Figure 8.1: A Conversation where Emotions Guide People's Thoughts.

upper case, the conversation continues in a happy way and arrives at a dance date when the conversation ends. When comes to the lower case, friend A sympathizes with B and asks what happened from the perspective of realizing B's sadness. This example shows that emotions usually guide people's internal states like intentions, and therefore words as well as the outcome of dialogue. Therefore, intelligent agents are expected to perceive users' emotions and reply emphatically using more pleasing expressions.

In this chapter, we aim to investigate how emotions influence other internal factors and semantics conveyed in the conversation. Specifically, we will use **intention** as a representative factor to explore, which is also another important factor that reflects people's thinking. In the aforementioned example, when A cares about why B is upset, he/she expresses the worry towards B instead of insisting on the details of the invitation. We hereby assume that emotion shapes the idea of intention, and both of these two factors mediate the response semantics. We use two *discrete* variables to capture the speaker's intention and emotion, and use the *continuous* variable to represent changes in the content level as similar to [189]. In order to incorporate these variables, we adopt the basis of variable encoder-decoder and design a novel hierarchical conditional model. Given an input along with history utterances, we transform them in context-aware representation, and firstly infer the interested variables in a hierarchy of emotion $\rightarrow$ intention $\rightarrow$ content, and then form the responses based on

the variable predictions. Such hierarchy in the prediction procedure captures inherits the dependency among emotion, intention and content, which for example, allows the conversation topics to be guided by the internal factors. In addition, since the intention now depends on emotion, it will facilitate much smarter and emotionally rich conversations. In this way, based on the overall context, the response semantics is expected to be more reasonable when considering the whole conversation context. The developed model is termed as **HINTE**, which stands for generating responses with **H**ierarchical **INT**ention and **E**motion prediction. We are also interested in examine the hypothesis we assume on the variables, and thus we implement several model variants to realize different relationships among the variables.

The remaining issue is how to efficiently train the developed HINTE to generate responses with desired properties. There are two difficulties. The first one is the vanishing latent variable problem that the decoder often bypasses the variable to be conditioned during generation, as identified in previous studies [20]. Even if the variables are aware to the decoder, it is still non-trivial to guarantee the variables are fully expressed in the generated responses. To remedy these two issues, we devise a adversarial learning approach as inspired by [61] to supervise the generation model on the variable-level. This brings us the benefit that the behavior monitoring on the variable-level is more effective than that conducted on the response level. The benefit is also verifies through comparison and ablation studies with other two adversarial approaches [292, 78]. To highlight, we conclude our contributions in below:

- We model two typical factors, emotion and intent for dialogue generation, and deploy a hierarchical conditional model to examine the effectiveness.

- We explore the relationships among emotion, intention and content in conversation modeling, hypothesize that emotion puts a high-level effect on intention, and investigate several model variants to validate it.

- We design a new adversarial learning objective, and empirically enhance the model performance, which has also been demonstrated beneficial through ablation studies.

The rest of this chapter is structured as follows. In Section 8.2, we give a brief survey on exiting hierarchical conversation models related to our work. In Section 8.3, we describe the technical details the proposed method on incorporating both emotion and intention information. In Section 8.4, we evaluate the model performances and deeply analyze the contribution of each component in the proposed approach. At last, we summarize this chapter in Section 8.5.

8.2 Related Work on Hierarchical Conversation Models

Due to the development of massive data and neural networks, researchers are trying to build conversational agents using generation-based methods. Because history dialogues usually provide a lot of information for conversation modeling, researchers have proposed a wide range of context-aware dialogue models. The easiest way is to use concatenation [126, 209, 279] and averaging [209, 223] to combine the historical utterances with the current utterance and feed them as a whole input to the model. A more sophisticated approach is to use a hierarchical encoder by treating the dialogue as a two-level sequence [208].

When modeling the structure of the conversation, [189] devises HRED with an additional variable to encourage response diversity. Afterwards, [195] provides each interlocutor with a speaker-aware encoder to improve VHRED, and then combines its hidden states to form a high-level dialogue context. [30] injects the variability of memory read through latent variables, and learns to make abstract high-level decisions in the dialogue tracking process. Their memory has been enhanced in the

hierarchy and updated to remember each utterance. Recent work in this area uses a global random variable that is conditioned on the speaker and finally affects the context [9]. Similar to [189, 195], our work is also based on the conditional variable framework and is novel in that we focus on the dependency among the variables. To the best of our knowledge, rarely has such dependency been investigated before.

8.3 Method

We develop a variational framework, HINTE, that is able to generate responses conditioned on the variables predicted in hierarchy. In particular, HINTE consists of the following parts: (1) History and Utterance Representation; (2) Hierarchical Intention and Emotion Prediction; (3) Conditional Response Generation.

We depict the designs of HINTE in Figure 8.2, where the lower block depicts the overview of HINTE. As shown, the proposed framework consists of three key modules. The most important module is the hierarchical variable prediction, which we expand in the upmost part with five typical relations among $\mathbf{z}_e$, $\mathbf{z}_i$ and $\mathbf{z}_c$. In specific, the three variables, i.e., emotion, intention and content, are demonstrated using different colors and shapes. Below, we will firstly give a picture of our framework by sketching three modules, and then describe the novel designs in the next subsections.

8.3.1 History and Utterance Representation

We aim to produce speaker-aware responses by capturing the speaker's emotion and intention states. To achieve it, we following [39, 46] and formulate emotion and intent using two discrete variables $\mathbf{z}_e$ and $\mathbf{z}_i$, respectively. In specific, the discrete variables are indexed by the pre-defined categories of emotion and intention variables. Denote that u_t and c_t are the utterance input and context representation at the time step t, respectively. The task is formally defined as generating response(s) given $\{c_t, \mathbf{z}_i^t, \mathbf{z}_e^t\}$. For better readability, we will omit the subscripts without harming the technical

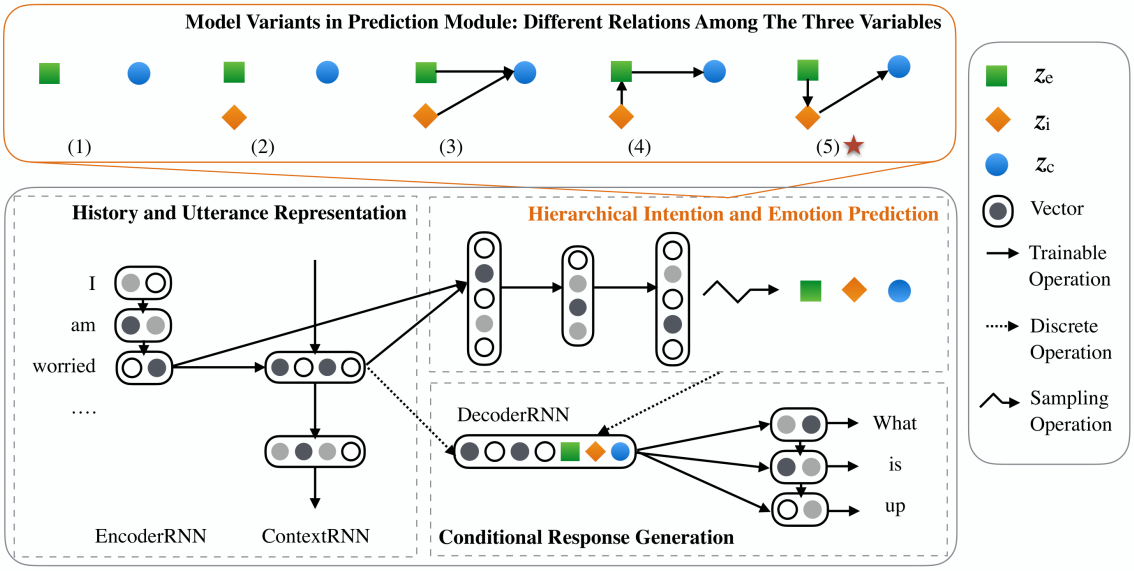

Figure 8.2: The Hierarchical Variational Generation Framework HINTE.

clarity.

Based on the architecture of VHRED [189], the history and current utterances are processed through:

$$\mathbf{u}_t = \text{EncoderRNN}(x_1, \cdots, x_{t-1}) \tag{8.1}$$

$$\mathbf{c}_t = \text{ContextRNN}(u_1, \cdots, u_t) \tag{8.2}$$

$$\mathbf{c}_t = f(\mathbf{u}_{t-1}, \mathbf{c}_{t-1}) \tag{8.3}$$

Based on the obtained representation, the decoder p_θ generates each word in the response by:

$$y_t \sim p_\theta(y_t|\mathbf{c}_t, y_1, \cdots y_{t-1})$$

To capture variation during decoding, there is a latent variable $\mathbf{z}_c$ injected into VHRED to influence the generation by:

$$y_t \sim p_\theta(y_t|\mathbf{z}_c, \mathbf{c}_t, y_1, \cdots y_{n-1}) \tag{8.4}$$

Initially, $\mathbf{z}_c$ was brought in for language modeling and one-sided sentence generation [20], which has been used to capture high-level information like themes,

emotions, styles, and other interpretable features [20, 189]. However, the ambiguity in $\mathbf{z}_c$ makes it difficult to capture precisely. In addition, if the model is able to capture other influencing factors and utilize them for understanding and generation, the semantics to be conveyed in the responses will be better learned.

Motivated by this, we propose to incorporate two more variables $\mathbf{z}_e$ and $\mathbf{z}_i$ and bring in the emotion and intention information in addition to the latent variable $\mathbf{z}_c$. We derive the three variables using the context representation obtained through EncoderRNN and ContextRNN. Notably, this work aim to explore the relationships among the variables. To achieve this, we establish a hierarchical predictive model, which allows us to infer the variables in turn by taking into account the hierarchy among them. We will describe the details in Section 8.3.2.

After the hierarchical prediction, the proposed framework HINTE is able to finally produce the speaker-aware responses by depending on the derived variables $\mathbf{z}_e$, $\mathbf{z}_i$, and $\mathbf{z}_c$. Formally, the decoder now generates a word by:

$$y_n \sim p_\theta(y_n|\mathbf{z}_i, \mathbf{z}_e, \mathbf{z}_c, \mathbf{c}_t, y_1, \cdots y_{n-1}) \tag{8.5}$$

Overall speaking, the proposed framework HINTE is equipped with a generation hierarchy to predict the latent variables one-by-one before generating the conditional responses. Because there are three latent variables together influence the generation performance, it is non-trivial to sufficiently learn the model. Hence, we further enhance the inference networks in the proposed framework with a novel adversarial learning objective, which will be presented in Section 8.3.3. As will be demonstrated by the experiments, the augmentation will not only benefit the model training but also facilitate the response generation and help HINTE to produce higher-quality responses.

8.3.2 Hierarchical Intention and Emotion Prediction

The primary contribution of the proposed network is the hierarchical prediction for the three latent variables. Using the context representations, we propose to infer the proper emotion and intention to be conveyed, and exploit them to form a speaker-aware response(s). Before generating the corresponding response, the model needs to infer the discrete and continuous variables. Note that the discrete variables are defined on the utterance-level, and the continuous variable is on the context-level. Therefore, there are potentially numerous combination of these variables in the auxiliary hierarchy when incorporating these three variables into the model.

Hierarchical Model

As indicated in the study [230], the discrete factor is usually at a higher level during the generation process when multiple variables exist. In addition, research works in theoretical psychology and communication have surmised and empirically shown that emotion and intention have impacts on the content of the response [39, 197]. To achieve this, it is possible to impose the discrete variables to be predicted:

$$\mathbf{z}_i \sim p(\mathbf{z}_i|\mathbf{u}_1, \cdots \mathbf{u}_t)$$

$$\mathbf{z}_e \sim p(\mathbf{z}_e|\mathbf{u}_1, \cdots \mathbf{u}_t)$$

$$\mathbf{z}_c \sim p(\mathbf{z}_c|\mathbf{z}_i, \mathbf{z}_e, \mathbf{u}_1, \cdots \mathbf{u}_t)$$

It is worth-noting that there are other options for how to model the relationship between emotion and intent. They can be predicted independently of each other. One may also rely on the other and predict conditionally. We choose the hierarchy based on the findings from the literature on emotional intelligence that emotions usually guide a person's thoughts and ultimately affect the outcome of the conversation [138]. As such, we hypothesizeethat emotions have an impact on intentions, and adopt a

three-level prediction hierarchy in the developed HINTE:

$$
\begin{aligned}
& p(\mathbf{z}_i, \mathbf{z}_e, \mathbf{z}_c | \mathbf{u}_1, \cdots \mathbf{u}_t) \\
=& p(\mathbf{z}_c | \mathbf{z}_i, \mathbf{z}_e, \mathbf{u}_1, \cdots \mathbf{u}_t) p(\mathbf{z}_i, \mathbf{z}_e | \mathbf{u}_1, \cdots \mathbf{u}_t) \\
=& p(\mathbf{z}_c | \mathbf{z}_i, \mathbf{z}_e, \mathbf{c}_t) p(\mathbf{z}_i | \mathbf{z}_e, \mathbf{c}_t) p(\mathbf{z}_e | \mathbf{c}_t)
\end{aligned}
$$

The three-level prediction procedure is illustrated in Figure 8.2 (5), which is: $\mathbf{z}_e \rightarrow \mathbf{z}_i \rightarrow \mathbf{z}_c$. Without loss of generality, we also propose several model variants and compare with them to validate the hierarchical hypothesis. To this end, we also illustrate the five main variants in the upmost part of Figure 8.2, where (1)(2) represent the prediction procedure with two or three independent variables, (3) models a two-level inference process, and (4)(5) stand for three-level hierarchy. It is clear that the assumption in (4) is totally contradicted to the hypothesis we adopt as the main design.

To obtain the final predictions of emotion and intention, we sample from the corresponding distributions $p(\mathbf{z}_e)$ and $p(\mathbf{z}_i)$, respectively. Since they are discrete variables, the predicted class is selected as the vector dimension with the largest prediction value. To incorporate the predicted variables into response generation, we then cast the discrete classes into one-hot embeddings, which are then combined together with the continuous variable to pass to the decoder. In this way, the proposed HINTE is able to respond by firstly predicting the latent variables in a hierarchical procedure $\mathbf{z}_e \rightarrow \mathbf{z}_i \rightarrow \mathbf{z}_c$, and then consider the predictions to form the variable-aware responses.

These three variables are predicted by approximating their posterior distributions using three inference networks, which are denoted as $q_\phi^e(\mathbf{z}_e | \mathbf{u}_1, \cdots \mathbf{u}_t)$, $q_\phi^i(\mathbf{z}_i | \mathbf{u}_1, \cdots \mathbf{u}_t)$, and $q_\phi^c(\mathbf{z}_c | \mathbf{z}_i, \mathbf{z}_e, \mathbf{u}_1, \cdots \mathbf{u}_t)$. Practically, they are Gaussian distributions:

$$q^e_\phi(\mathbf{z}_e|\mathbf{u}_1, \cdots \mathbf{u}_t) = \text{Multi}(\mathbf{o}^e_t) = \text{softmax}(\mathbf{W}^e\mathbf{o}^e_t)$$

$$\mathbf{o}^e_t = \text{MLP}^e(\mathbf{u}_t, \mathbf{h}^C_t)$$

$$q^i_\phi(\mathbf{z}_i|\mathbf{z}_e, \mathbf{u}_1, \cdots \mathbf{u}_t) = \text{Multi}(\mathbf{o}^i_t) = \text{softmax}(\mathbf{W}^i\mathbf{o}^i_t)$$

$$\mathbf{o}^i_t = \text{MLP}^i(\mathbf{u}_t, \mathbf{h}^C_t, \mathbf{z}_e)$$

where $\mathbf{o}_t$ is an integrated representations for hidden inputs, and Multi() stands for a feed-forward neural network.

Variational Bounds

During inference, the latent variables are inferred by maximizing the variational bounds. For readability, the subscript for c_t is omitted, and $\mathbf{z}_d$ is introduced to represent either $\mathbf{z}_e$ or $\mathbf{z}_i$. Also, we use q^d to refer to q^e or q^i. When the labels for the discrete variables, i.e., $\mathbf{z}_d$ are unseen, that is, $\mathcal{X} = \{W_i\}_{i=1}^K$. In this case, for any $W = (w_1, w_2, \cdots w_N) \in \mathcal{X}$, the variational bound for the unsupervised setting is derived as follows:

$$\begin{aligned}
\log p(w_1, \cdots, w_N) \geqslant \sum_{n=1}^{N} & -\text{KL}(q^d_\phi||p(\mathbf{z}_d|c)) \\
& - \text{KL}(q^c_\phi||p(\mathbf{z}_c|\mathbf{z}_d, c)) \\
& + \mathbb{E}_q[\log p(w_n|\mathbf{z}_c, \mathbf{z}_d, c)] \\
& := -V_{\text{un}}
\end{aligned} \tag{8.6}$$

However, the models trained with the above learning objective are often observed unstable because of the high sample variance [251]. In order to mitigate this problem, we adopt supervised learning approaches to provide fine-grained signals for the model learning. When we are able to access the labels of $\mathbf{z}_d$, i.e., $(\mathcal{X}, \mathcal{Z}) = \{(W_i, Z_i)\}_{i=1}^K$, we put the supervisions $Z = (\mathbf{z}^1_d, \mathbf{z}^2_d, \cdots \mathbf{z}^N_d) \in \mathcal{Z}$ on $W = (w_1, w_2, \cdots w_N) \in \mathcal{X}$. Then

we derive the variational bound in the supervised setting as:

$$\begin{aligned}\log p(W, Z) \geqslant \sum_{n=1}^{N} & -\mathrm{KL}(q^c || p(\mathbf{z}_c|\mathbf{z}_d, c)) \\ & + \mathbb{E}_{q^c}[\log p(w_n|\mathbf{z}_d, \mathbf{z}_c, c)] \\ & + \log p(\mathbf{z}_d|c) \\ := & - V_{\text{sup}} \end{aligned} \tag{8.7}$$

According to the variational bounds given above, we are now able to derive the learning objective for both the unsupervised the supervised settings. Formally, we denote the objectives as $L_{\text{sup}} := \mathbb{E}_{W,Z\sim(\mathcal{X},\mathcal{Z})}[V_{\text{sup}}(W, Z)]$ for supervised setting, and $L_{\text{un}} := \mathbb{E}_{W\sim\mathcal{X}}[V_{\text{un}}(W)]$ for the unsupervised setting.

8.3.3 Adversarial-augmented Inference Learning

We maximize the objective of log-likelihood when training the model. Unfortunately, solely relying on the objective itself is insufficient to ensure the generated responses resemble the desired properties. Even if the content in the generated response is regarded as emotional and intentional by the machine, they might not be sensible for humans. An example case is that the machine may predict a generated response containing painful words as the emotion category of happiness. This inconsistency is partially attributed to the free generation process that the decoder will get no penalty even if the generated responses are ridiculous to the predictions. Considering the architecture of decoder is autoregressive, i.e., RNNs, the situation is even exacerbated. It is because that RNNs will impose stronger conditional constraints between the neighboring words, as revealed by [20]. As a result, the information brought by the latent variables z will get lost and turn weak during model learning.

There are two additional issues we observe that exaggerate the worse situation. When generating words, the decoder will neglect the information predicted by the la-

tent variables, the similar vanishing variable problem as in [20]. Ideally, the decoder is expected to concentrate on minimizing the KL-divergence term meanwhile keeping the reconstruction loss. When the decoder is able to directly access the encoder, however, the learning procedure will be deviated since it easily accesses the input information. Theoretically, if $q_\phi(\mathbf{y}|\mathbf{z})$ can perceive $\mathbf{X}$ (i.e., the information on the encoder side), then $q_\phi(\mathbf{y}|\mathbf{z})$ may be learned almost identical as $q_\phi(\mathbf{y}|\mathbf{x})$. In addition, the latent variables would not acquire sufficient information during training and thus are difficult to learn well if they are not regularized. The undesired phenomena are resulted from the lack of supervision on the conditional generation behavior. To alleviate these problems, we deploy the idea of adversarial learning [61] to monitor the variable learning. Intuitively, given a real response, humans are able to deduce its underlying emotion and intention. This implies that a high-quality generated response should to be consistent with the latent predictions, i.e., the predicted categories of emotion and intention.

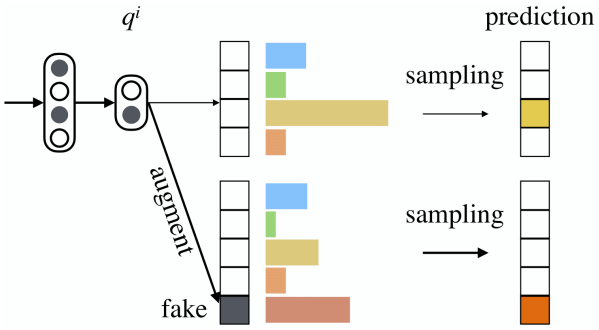

Figure 8.3: The Augmented and Original Inference Networks.

Inspired from this, we augment the inference networks and "transform" it into discriminators. In particular, an extra class is appended to the output of the inference networks, which represent the "fake" category. For illustrative understanding, we take the intention inference network q^i as a case, as shown in Figure 8.3. The color bars are depicted to stand for the weight values output from the inference networks,

which means how likely each category is for the discrete variables, as predicted by the network. The selected variable category will be set as 1 after sampling, whereas others as 0. The emotion inference network q^e is augmented in the same way. Denoting the space of all possible intention classes as Z_i, we define the augmented latent factor $\mathbf{z}_i^* \in Z_i \cup \{\mathbf{F}\}$ where $\{\mathbf{F}\}$ represents the generated (fake) sentence. Now we have $q^i(\mathbf{z}_i^*|c, w_1, \cdots, w_n)$. The objective for the new augmented q^i is:

$$L_{q^i} := \mathbb{E}_{W,Z\sim(\mathcal{X},\mathcal{Z})}[\sum_{n=1}^{N} \lambda_i(\mathbf{z}_i) \log q_\phi^i(\mathbf{z}_i|c, w_n)$$

$$+\lambda_i(\mathbf{F}) \sum_{n=1}^{N} \log q_\phi^i(\mathbf{z}_i^* = \mathbf{F}|c, \tilde{w}_n(\mathbf{z}_i))] \tag{8.8}$$

where λ_i are the class weights for tackling unbalanced class labels. By augmented with an extra class, the inference networks q^i and q^e now perform like "discriminators" to have more awareness of the variable sensibility. This will in turn aid in the learning of the generator, a.k.a. response generation. When the discrete variables are discriminated as "fake", the DecoderRNN should be penalized. To achieve this, we derive an adversarial loss as:

$$V_{adv}(w_1, \cdots, w_n) = \mathbb{E}_{\mathbf{z}_i}[\sum_{n=1}^{N} \log q^i(\mathbf{z}_i|\tilde{w}_n(\mathbf{z}_i), c)] \tag{8.9}$$

and defined by $L_{adv} := \mathbb{E}_{W\sim\mathcal{X}}[V_{adv}(W)]$.

We then add these two adversarial losses derived from q^i and q^e to the variational objective. The combination of the losses will together influence the decoding procedure, whose objective now turns into $L_g = L_{\text{sup}} + \lambda_{adv} L_{adv}$ or $L_g = L_{\text{un}} + \lambda_{adv} L_{adv}$, where λ_{adv} is a weight tuning the constraints of the adversarial learning, and L_{sup} or L_{un} is the variational objective as defined in Eq. 8.7 and 8.6. The objective for the continuous encoder q^c remains the same. It is trained jointly with the decoder, using normal back-propagation, on both labeled and unlabeled data. Finally, we generate

responses $\tilde{w}_n(\mathbf{z}_i, \mathbf{z}_e)$ by replacing each $\arg\max(x)$ word choice with $\text{softmax}(x/\alpha)$, where α is the temperature parameter.

It is worth-noting that our way to apply adversarial learning is different from previous methods that applied adversarial learning on the response-level with an extra independent discriminator. Rather, we devise the inference networks, i.e., q^i and q^e, and train them to discern the variable-level imperfections, which bring in at least two-fold benefits: (1) After augmentation, q^i and q^e are able to perceive the response inconsistency based on the integration of variable predictions and conversation history, which also deduces the response generator to become stronger in accordance; (2) Because the representations adopted in q^i and q^e are shared with other components in the proposed HINTE, it can be regarded as a regularizer to aid in representation learning. Since our novel design directly regulates the conditional behavior on the variable level, the way of applying adversarial learning proposed in this chapter is more effective and efficient, which is verified through extensive experiments in the following section.

8.4 Experiments

We design three groups of experiments to examine the proposed approach along with each novel design. In the first group, we compare with other state-of-the-art models and demonstrate the powerfulness of the proposed HINTE (Section 8.4.3). In the second group, nine model variants are implemented to validate the assumption that emotion shapes the behavior of intention (Section 8.4.4). The third group of experiments is set to examine the advantage and novelty of the newly devised adversarial learning objective through comparing with similar methods (Section 8.4.4). We also conduct ablated studies at last.

Table 8.1: Statistics of Corpus DailyDialog.

Total Number of Conversation	13,118
Average Speaker Turns Per Conversation	7.9
Average Number of Tokens Per Conversation	114.7
Average Number of Tokens Per Utterance	14.6

8.4.1 Dataset

We adopt the dataset DailyDialog [113] to validate the effectiveness of the proposed HINTE. DailyDialog is a publicly available conversation corpus involving 13,118 multi-turn conversations. It is attractive in that each utterance is annotated with both emotion and intention class with three experts. As far as we know, DailyDialog is the only one chit-chat corpus that is tagged with both emotion and intention labels manually. This makes it the only one suitable to assess the proposed hierarchical response generation model. We follow the official divisions for training/validation/testing as in [113]. Please refer to chapter 4 for more on the annotation details,

8.4.2 Experimental Setup

Compared Models

In order to investigate whether the proposed HINTE is effective on open-domain response generation, we compare it with several state-of-the-art models:

- ENC2DEC-ATTN [8]: This is the vanilla Seq2Seq model equipped with the attention mechanism.
- HRED [208]: This state-of-the-art model incorporates history utterances, where a conversation-level ContextRNN is on the top the word-level utteranceRNN.
- TRANSFORMER [228]: This is a newly established state-of-the-art text genera-

tion model. We adopt the implementation here.[1]

- VHRED [189]: This model enhances the capability of hierarchical conversational models by including a latent variable to allow more variations in response generation.

- SPHRED [195]: This model is proposed and described in chapter 4, where the responses are produced base on the speaker-aware context representation as well as a latent variable guided by an external label.

We use TensorFlow [1] to implement all the models, and concatenate the one-vector labels into the decoding vectors of all the compared models. The vocabulary is restricted to 25,000 words, where the OOV words are mapped to UNK. We adopt Google's 300-dimentional word embeddings,[2] and normalize the embeddings during training. All low-level word-level encoders are defined as 1-layer GRUs with 512 hidden neurons. The high-level context-level encoders in HRED [208] and VHRED [189] are both 1-layer bidirectional GRUs with 1,024 hidden units. We set the minibatch size to 128, and set the learning rate as a fixed number of 0.0002. Adam optimizer [92] is adopted for model training.

Evaluation Metrics

We adopt two commonly used automatic scores to assess the model performances, i.e., BLEU-n [161] and Distinct-n [101]. Nevertheless, according to the research [119, 156], N-gram based scores like BLEU usually are often inconsistent with human judgments when assessing dialogue models. In order to complement the evaluation, we randomly pick up 100 test samples and perform manual assessments on them. We train three annotators with linguistic background and send the samples to them. To be fair,

[1] `https://github.com/EternalFeather/Transformer-in-generating-dialogue`

[2] `https://code.google.com/archive/p/word2vec/`

Table 8.2: Main Experimental Results.

	Latent Variables	BLEU-1	BLEU-2	BLEU-3	Dist-1	Rel.	Flu.	Divers.	Appr.
Enc2Dec-Attn	✗	0.435	0.208	0.017	0.038	4.24	3.66	3.98	3.52
HRED	✗	0.412	0.176	0.020	0.053	3.62	3.81	4.11	3.44
Transformer	✗	0.458	0.210	0.029	0.052	3.32	2.79	3.83	3.10
VHRED	✓	0.418	0.181	0.018	0.058	1.94	2.15	2.32	3.48
SPHRED	✓	0.443	0.198	0.021	0.062	1.85	1.59	2.17	2.04
HINTE	✓	**0.471**	**0.214**	**0.023**	**0.075**	**1.18**	**1.34**	**1.51**	**1.62**
HINTE (40%)	✓	0.429	0.183	0.021	0.62	-	-	-	-
HINTE (20%)	✓	0.380	0.161	0.021	0.62	-	-	-	-
HINTE (10%)	✓	0.271	0.103	0.003	0.19	-	-	-	-

these annotators have no idea about which model the test response belongs to. The annotators are asked to consider the following 4 aspects when rating the generated response [106]: {Relevance, Fluency, Diversity and Appropriateness} as before. For the details of the annotation criteria, please refer to Chapter 4.

Another thing to note is that, the annotators are notified to give the same rank, when the responses to be compared share a large piece of overlapping with each other. In general cases, the annotator will give a higher rank to a better response. The comparison results are present in Table 8.2.

8.4.3 Performance Evaluation

We begin with studying the importance of latent variable. According to the majority of scores across Table 8.2, the chatbots in the first block (first three rows) perform the worst than the models in the second block (last three rows). Although TRANSFORMER beats VHRED and SPHRED in BLEU-1 and BLEU-2 scores, it obtains a poor Distinct-1 score. This is because the responses generated from Enc2Dec-Attn and Transformer usually contain a series of common words, such as "I don't know", "thank you", "I like it", etc. These common but useless words greatly contribute to Transformer's high BLEU scores. On the other hand, the responses generated by HRED, VHRED and SPHRED are more informative, which can be concluded from their satisfactory Dist-1 scores. The contradiction between BLEU scores and human judgment is consistent with the suggestion from [120, 156] that, BLEU-n is not a good indicator for evaluating dialogue models.

We then look deeper for the reasons for the model failures. The problem of "safe response" may be due to the one-to-many relationship between a given input utterance and its possible multiple proper responses. By using more information such as intent and emotion, latent variable based models can reduce the number of possible responses by learning a more compact and precise representation of the

response to be generated. As a result, those responses will be preferred if they are compatible with the emotion and intention in the conversation context, a.k.a. input utterance(s). Since Enc2Dec-Attn and HRED only utilize the latent variable in the decoder without considering them in context modeling, their experimental results prove that, it is instrumental to incorporate influential variables using effective approaches.

Since all the other three models employ latent variables in context modeling, we compare them to find out which model can capture the variables more effectively. By checking the last three rows in Table 8.2, we can see the differences among VHRED, SPHRED and HINTE. Although similar in the way of modeling history information, the proposed HINTE differs from VHRED and SPHRED in how it uses intent and emotional information. Note that VHRED and SPHRED simply concatenate intent and emotion information in a single, plain vector, and the functions of variables are independent. Therefore, it is difficult for these two models to distinguish information from different variables in the learning process. In contrast, our HINTE exploits these variables in a hierarchical manner to allow emotion and intention to adjust content variables.

It is obvious that HINTE and SPHRED yields the best performances. By comparing they two, we can find that both of them allow the external label to guide the variable learning. However, it is difficult to train a variational model, because variational inferences often suffer from high sample variance. In the early stages, the training process will be unstable and the produced subtle samples will prohibit the chabots from receiving reliable learning signals, resulting in a large decrease on the scores. One way to alleviate the issue is to apply semi-supervised learning [93] to aid in variational model learning [78, 251]. Therefore, we also investigate the performance of HINTE based on the scheme of semi-supervised learning. To use different scales of labelled data, we randomly remove certain ratios of data labels as

Table 8.3: The Compared Model Variants.

	#Variable	#Level	Unshared	Adv	Description
HINTE-EC	2	1	U	-	$\mathbf{z}_e$, $\mathbf{z}_c$, independent, as depicted in Fig. 8.2 (1).
HINTE-IC	2	1	U	-	$\mathbf{z}_i$, $\mathbf{z}_c$, independent, similar with Fig. 8.2 (1).
HINTE-I	3	1	U	-	$\mathbf{z}_e$,$\mathbf{z}_i$,$\mathbf{z}_c$,independent, as illustrated in Fig. 8.2 (2).
HINTE-E-C	2	2	U	-	$\mathbf{z}_e \rightarrow \mathbf{z}_c$, conditional
HINTE-EI-C(S)	3	2	S	-	$\mathbf{z}_e \rightarrow \mathbf{z}_c$,$\mathbf{z}_i \rightarrow \mathbf{z}_c$,conditional, shared, as depicted in Fig. 8.2 (3)
HINTE-EI-C(U)	3	2	U	-	$\mathbf{z}_e \rightarrow \mathbf{z}_c$,$\mathbf{z}_i \rightarrow \mathbf{z}_c$,conditional,unshared, as illustrated in Fig. 8.2 (3)
HINTE-I-E-C	3	3	U	-	$\mathbf{z}_i \rightarrow \mathbf{z}_e \rightarrow \mathbf{z}_c$, hierarchical, controversial, as depicted in Fig. 8.2 (4)
HINTE-E-I-C	3	3	U	-	$\mathbf{z}_e \rightarrow \mathbf{z}_i \rightarrow \mathbf{z}_c$, hierarchical, as depicted in Fig. 8.2 (5).
HINTE	3	3	U	+	$\mathbf{z}_e \rightarrow \mathbf{z}_i \rightarrow \mathbf{z}_c$, hierarchical, adversarial

unsupervised set, and train HINTE under the paradigm of semi-supervised learning, which is equal to Eq.(8) and Eq.(9) as present before. From the last block in Table 8.2, we can observe that as the ratio decreases, the model also degrades. It is also worth-noting that the effect of variable labels is more significant when the ratio of training data is under 20%. Despite the limited improvements when extra 20% data are available, HINTE (40%) still achieves competitive results to SPHRED (100%).

Generally, our HINTE is more superb as indicated by all evaluation metrics. We will show some case studies in the next subsection.

8.4.4 Analysis

Hypothesis Validation and Ablated Study

In this subsection, we focus on studying whether the novel designs in our proposal are useful: (1) the prediction hierarchy of emotion $\rightarrow$ intention $\rightarrow$ content; (2) the adversarial objective applied on the variable-level. We implement and compare with three groups and in total nine model variants, as list in Table 8.3. To evaluate nine model variants, we first run automatic assessments and then make a little change for the human judgements. Because now there are nine models to be compared for each annotator, it is hard for them to rank from the best to the worst (1st to the ninth). Therefore, we modify the human judgment criteria to make the annotators rate the responses according to the four aspects. The comparison results are given in Table 8.4.3.

It is observable that the independent models in the first block, namely HINTE-EC, HINTE-IC, HINTE-EIC, are the worst variants. Among they three, HINTE-EIC demonstrates a negligible improvement over the other two models. In spite of it, HINTE-EIC is still far from competitive with the other two blocks of model variants. The comparison between the blocks affirms the benefit of modeling high-level discrete variables over response generation. Moreover, regarding to the distinctions between

Table 8.4: Hypothesis Validation Results.

	BLEU-1	BLEU-2	BLEU-3	Dist-1	Rel.	Flu.	Divers.	Appr.
HINTE-EC	0.396	0.174	0.019	0.048	1.96	1.67	1.81	2.02
HINTE-IC	0.379	0.152	0.018	0.052	2.08	1.74	1.81	2.02
HINTE-EIC	0.418	0.181	0.018	0.055	2.14	1.72	1.88	2.08
HINTE-E-C	0.431	0.189	0.018	0.040	2.07	1.79	1.78	2.04
HINTE-EI-C(S)	0.393	0.174	0.018	0.039	1.68	1.99	1.74	1.97
HINTE-EI-C(U)	0.437	0.193	0.020	0.054	2.62	1.94	1.90	2.12
HINTE-I-E-C	0.443	0.188	0.018	0.052	2.32	1.99	1.92	2.04
HINTE-E-I-C	0.464	**0.224**	**0.023**	0.074	2.52	2.10	2.24	2.10
HINTE	**0.471**	0.214	**0.023**	**0.075**	**2.67**	**2.16**	**2.28**	**2.21**

HINTE-EC and HINTE-IC, and comparing them to HINTE-I, it is promising to see that intention and emotion are complementary to each other.

We then shift attention to examine the performances among the second block of models. It is interesting that HINTE-EI-(S) and HINTE-EI-C(U) yield distinguishing performances even though they two share the "same" two-level hierarchy of prediction. According to the indicators, the conditional model HINTE-EI-(S) even falls behind the independent counterpart HINTE-EIC. This result implies that it is useless and even notorious to predict the different discrete variables using the same feed-forward inference networks, because the inference networks will struggle to predict precisely based on the entangled representations.

As surmised before, HINTE-E-I-C and HINTE surpasses the other variants owing to their hierarchical conditional procedure. Their overwhelming performances validate the hypothesis that content is guided by emotions and intentions in a hierarchical manner. By comparing the model performances between the second and the third block, we are able to draw conclusions that by modeling the emotion's influence on the intention, both HINTE-E-I-C and HINTE are capable of utilizing information more effectively to predict the influential factors and in turn better bias the response generation. It is intuitive that humans will finish a dialogue when being enraged by the other speaker in the conversation. As argued by [138], when experiencing a certain kind of emotion, humans usually are evoked with special reactions like terminating the chats with others. Similar findings are also observed in the studies on psychology and communication theory [148, 163, 197].

Study on Variable Inference with Adversarial Learning

The difference between HINTE-E-I-C and HINTE in Table 8.4.3 is evident that introducing adversarial learning is of benefit for the variable prediction in response generation. In specific, the devised adversarial learning is applied on the variable-

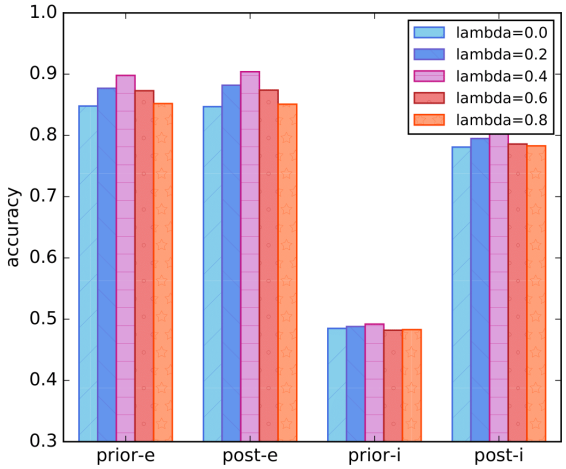

Figure 8.4: The Influence of λ_{adv} on Adversarial Learning Objective.

level, which assists the representation learning. To quantitatively measure to what extend the adversarial learning benefit the predictions, we explore the effect of the coefficient λ_{adv}. Based on different value of λ_{adv}, we compare the priors (i.e., $p(\mathbf{z})$) and the posteriors (i.e., $q(\mathbf{z}|\mathbf{x})$) of the discrete variables with the real labels. and illustrate the result in Figure 8.4. The best performance is achieved when λ_{adv} is increased to 0.4. Neither a smaller nor a larger value is desired, because a smaller λ_{adv} will not bring in enough regulation from the adversarial monitoring, and a larger λ_{adv} will on the contrary make the learning procedure dominate by the adversarial gradients.

Notably, the prior distributions of intent and emotion are quite different, as shown in Figure 8.4. The reason is that these two variables are distributed distinguishedly in the original dataset. As mentioned in Chapter 4 [113], the intent factor is labelled with balanced category distribution, while the category distribution of emotion is totally uneven. This greatly influences the learning of these two variables in HINTE. As discussed in [170], though there are multiple categorization and dimensional taxonomies available, it is challenging to decide which annotation criteria is the most suitable. Although a simple categorization has obvious shortcomings, it is also non-

trivial for annotators to label the dataset using complex emotion models with a satisfactory agreement.

Since the other novel design in the developed HINTE is the variable-level adversarial objective, we then investigate its distinction and contribution by comparing it with the following approaches:

- HINTE-E-I-C: This is a variant of HINTE, as introduced in the previous subsection. Since this vanilla model is not equipped with adversarial learning, we introduce it as a baseline model.

- HINTE-R [292]: The first design we compare is the original idea proposed in [61], where a free discriminator D is deployed to decide whether the test sample is a true example from the dataset, or a synthesized one produced by the generator G. To make a reasonable comparison, we perform two adaptions. Firstly, we replace the adversarial loss in HINTE with a classification one classified by an external convolutional neural network based discriminator D. Secondly, we adapt the generator G in HINTE, i.e., the response decoder, to be trained via REINFORCE algorithm [256] following [292], and the generator objective becomes as $L_g = \sum_t D(y_t) * \log p(y_t|y_{<t})$. While the proposal in HINTE is conducted on the variable-level, this compared model builds up the adversarial game on the response-level, and is thus denoted as HINTE-R.

- HINTE-C [78]: The second model to be compared is more similar to our proposal. It is introduced in [78] in order to also bias the variable in text generation with desired style like sentiment. Distinct from our proposal, the discriminator in this compared model is also disentangled from the generator.

Note that the compared models are equipped with the same response generators. This kind of control experiment makes us focus on the effect of their distinguishing

Table 8.5: Comparison of Sampling and Parameter Choices.

Sampling	α	Perplexity	BLEU-2	Posterior-emo.	Posterior-intent
deterministic	0.1	130.9	0.048	0.45	0.53
deterministic	1	127.2	0.039	0.49	0.48
stochastic	0.1	104.6	0.088	0.74	0.71
stochastic	1→0	66.0	0.214	0.90	0.82

parts, that is, how they utilize the idea of adversarial learning in the "disciminator" part and how about the effects. To quantitatively measure the model performances, we adopt perplexity (PPL), KL cost, BLEU-n and Distinct-n scores, as well as prediction accuracy. An ideal response decoder is expected to have low PPL, non-trivial KL costs, and accurate variable predictions. The results in Table **??** strongly supports that our design for variable-level adversarial learning is of great benefit.

Studies for Variable and Word Sampling

In our method, there are two technical designs that also have impacts on response diversity: the sampling method of latent variables and the sampling parameters of word decoding. These variables can be obtained deterministically using the argmax operation, instead of randomly sampling z_i and z_e. Another influential parameter is the temperature α during word decoding. To obtain a deeper investigation, we design several sets of control experiments with respect to these two factors.

The comparison results are clearly revealed in Table 8.5 by the Perplexity scores of the first three rows against those in the last rows. The higher perplexity achieved by the deterministic variable models indicate that the deterministic models often struggle to converge due to the sharper learning signals acquired from the argmax operations. In other words, the stochastic models with sampling methods are more sufficiently trained when comparing with their deterministic counterparts. Among the two stochastic variants we compare, the one trained with a dynamic strategy of

the temperature achieves a significantly better performance. In specific, the value of the temperature is set to be a large number when the training starts and is modestly decreased to a lower number (almost zero) when approaching the ending point. Such dynamic strategy facilitates to induce the model at the beginning stage to be more diverse, and guarantee the prediction to be more reliable at the end.

8.5 Chapter Summary

We study context-level coherence by considering two conversation factors, emotion and intention, when modeling open-domain conversation. Innovatatively, we hypothesize that these two factors are not independent and we investigate the dependency among the factors by proposing a hierarchical conditional model, HINTE. In specific, we model an effect of emotion on top of the factor intention, and validate the hypothesis through extensive experiments. The proposed model HINTE is scalable because it is very straightforward to allow extra information modeling into it.

Chapter 9

Conclusions and Future Work

In the past decade, mobile devices has brought revolutionary changes on the way that information propagates among individuals. The evolution of messaging applications is now in full swing including WhatsApp, Slack, Chinese Wechat and their analogs. For example, both Facebook Messenger and Wechat have more than 1.2 billion monthly users from a wide range of age groups. With the spread of messengers, virtual conversational agents for assisting and accompanying human users are becoming increasingly in demand. For solving tasks, virtual assistants like Siri can fulfill information seeking need, hotel booking, check-in for a flight and so on. For social needs, chatbots like Xiaoice aim to build amicable bonds with users through entertainments and chit-chats. All these related conversational agents require good understanding of user needs and proper reactions towards users. This presents unprecedented challenges and opportunities for researches on developing conversational agents, driving many researchers, in recent years, to study the dialogues on messaging platforms and particularly focus on the problems of response retrieval and response generation, which study how to properly respond to the user based on the conversation context.

In this thesis, we comprehensively study the problem of response generation for open-domain social chatbots. Since conversation consists of multiple turns of ut-

terances, we tackle three-level of response coherence by proposing three important sub-problems, i.e., (1) how to exploit extra information in a limited design to improve utterance-level response coherence; (2) how to model the information dependencies in multiple turns to improve conversation-level coherence; and (3) how to capture the interactions between extra information and utterances to understand conversation context in a holistic view. Based on the neural encoder-decoder architecture, we proposed a series of models to utilize three kinds of extra information, including background knowledge, emotion and intention, to address these three problems. Compared with state-of-the-art studies, our proposed models consistently obtain significant improvements on response informativeness and response coherence.

9.1 Summary of Contributions

The following sections summarize the contributions of this thesis according to the information we explore.

9.1.1 Knowledge-aware Models

- Different from previous knowledge-grounded chatbots only injecting knowledge into response generation, we regard the necessities of knowledge in both context representation and response generation, and develop a chatbot MIKE to utilize KB attributes and entities in their own ways.

- We are the first to model conversation flow using meta-path information, and propose a meta-path augmented chatbot MOCHA to generate responses that are coherent to the conversation flow.

- The experimental results on two large-scale datasets demonstrate that it is more effective to incorporate structural knowledge than unstructured knowledge into

Seq2Seq models, and the proposed MIKE and MOCHA significantly outperform other state-of-the-art models.

- The further ablation studies verify that response coherence is remarkably improved because of the utilization of attributes, entities and meta-paths.

9.1.2 Emotion-aware Models

- We first propose a conditional variational model, SPHRED, which models the states of two speakers separately and learns to generate responses based on the predicted emotion.

- We then curate a conversation corpus, DAILYDIALOG, which exhibits a variety of natural human communication phenomenon. In particular, conversations are emotion-rich and manually labelled based on the "BigSix" emotion theory. The dataset has been included in the popular huggingface NLP platform as a benchmark conversation dataset.[1]

- The experimental results on the developed dataset demonstrate the significance of emotion for both retrieval-based and generation-based approaches, and show the effectiveness of the proposed framework, SPHRED. Meanwhile, it is also flexible to be applied to response generation controlled by any other kind of information.

- The further analysis reveals that the key of SPHRED is the success guidance on latent variable from emotion information. It emphasizes the importance of variable learning in controlled response generation, and inspires our later work on hierarchical variable conversation models.

[1] `https://github.com/huggingface/datasets/pull/556`

9.1.3 Context-aware Models

- While traditional context-aware models only model history utterances, we argue that conversation context also includes information like background knowledge and emotion. Most importantly, it is critical to model conversation by considering all the information in the conversation context together.

- We explore the effect of history utterances on entity reasoning when generating responses, and design a chatbot, CHEER, to achieve both social coherence and individual coherence.

- Unlike previous studies, which learn the representations of knowledge and utterances separately, we model external knowledge and utterances with a unified context graph, and develop, CGE, to allow the chatbot to have holistic understanding of conversation.

- We are the first to explore the dependencies of emotion and intention in open-domain response generation, and develop an adversarial-enhanced hierarchical model, HINTE, to firstly predict emotion, then intention, and lastly generate the response based on the predictions.

- On the benchmark datasets, we demonstrate the effectiveness of the proposed models, verify the hypothesis that emotion influences intention in daily communication. The experimental results show the importance of holistic context modeling for open-domain response coherence.

9.2 Future Work

At last, we point out the following potential directions that can further extend our previous work.

- In Chapters 3, 5, 6 and 7, we develop knowledge-grounded models and exploit the structure of knowledge like attribute and meta-path to improve response coherence. Despite the improvement, the chatbot equipped with external knowledge bases (KBs) have some shortcomings. Firstly, the performance of these chatbots heavily depend on the coverage and the accuracy of external KB, as we analyzed in Chapter 3. Secondly, the majority of KBs consist of certain factual knowledge from a specific domain, which is only a small fraction of our world knowledge. On the other hand, pretrained language models (PLMs) [40, 177] have been demonstrated powerful for various NLP tasks due to its ability to learn implicit knowledge inherited in the large-scale unlabeled corpora. Hence, it is promising but non-trivial to leverage PLMs to improve conversation models. Because standard PLMs only accepts a single sequence as input, current approaches utilizing PLMs for dialogues often simply concatenate the input dialogue history and the output response in fine-tuning stage [22] or duplicate PLMs in the encoder or the decoder side [60, 303]. However, as demonstrated in our thesis, it is necessary to incorporate multiple sources of information when modeling dialogues. Therefore, the first challenge is how to exploit the power of PLMs by fully leveraging the contextual information for dialogue models. Secondly, PLMs often set constraints on the maximum number of input tokens, and thus hinders the utilization of longer conversation history and external knowledge. It is necessary to devise efficient protocol to bypass the information bottleneck of PLMs. To this end, we plan to endow chatbots with knowledge from both external KB and PLMs in the future to tackle the aforementioned two research problems.

- Another potential direction of future work is to improve the sense of empathy of emotion-aware chatbot and endow it with the ability of pacifying users. In

Chapters 4 and 8, we develop emotion-aware chatbots which is only able to perceive coarse-grained user emotions and generate emotion-aware responses without any communication strategy. This is far from an emphatic companion as we human friends or counseling psychologists can do. Very recently, there are already some noteworthy advancements in the field of empathetic chatbots to accompany users or treat mental illnesses. For example, Woebot [48] uses methods from Cognitive Behavioural Therapy (CBT) to help people feel grounded during this unprecedented anxiety-provoking COVID-19 time. This offers evidence that intelligent chatbots can serve as a cost-effective and accessible therapeutic agent. Although not designed to appropriate the role of a trained therapist, integrative psychological AI emerges as a feasible option for delivering support. Therefore, it is interesting to improve the emotion-aware chatbots in Chapters 4 and 8 by building up a series of counseling abilities. When interacting with users, the chatbots should become more strategic to guide user express themselves more, and to pacify users' towards a calm or positive emotion. To achieve this, we plan to detect not only users' emotions but also the causes behind the emotions. The emotion cause will be helpful to form the response strategy. Also, we will design a set of communication intentions with respect to counseling behavior, and learn multi-turn counseling strategies to elicit and appease user emotions.

- Moreover, reasoning is also a crucial direction for future research on conversation modeling. On the one hand, there are a variety of information in conversation context, and it is a comprehensive decision when generating responses given multiple sources of information. While previous knowledge-grounded chatbots often rely on attention mechanism to select knowledge based on a plain similarity score, recent work has proposed reinforcement learning based

and graph network based methods to conduct reasoning more structurally. The key idea of recent methods is to formulate the information as graph nodes and incorporate the dependencies among the information as graph edges, which is similar to the context graph we define in Chapter 7. The distinction, however, lies in the reasoning part. Rather than using attention scores, some researchers adopt reinforcement learning to traverse over the graph and reach at the node to be selected. We can also apply this idea under our current framework to conduct entity reasoning over the context graph using reinforcement learning techniques. On the other hand, conversation is a dynamic process through multiple turns of information exchange. Therefore, how to reason over the turn-taking with newly involved information is also an interesting problem. Overall, how to effectively reason over the information graph and handle the dynamics of the conversation is a challenging problem to be solved.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[3] Rami Al-Rfou', Marc Pickett, Javier Snaider, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Conversational contextual cues: The case of personalization and history for response ranking. *CoRR*, abs/1606.00372, 2016.

[4] Dilafruz Amanova, Volha Petukhova, and Dietrich Klakow. Creating annotated dialogue resources: Cross-domain dialogue act classification. In *LREC*, 2016.

[5] Nabiha Asghar, P. Poupart, J. Hoey, Xin Jiang, and Lili Mou. Affective neural response generation. *ArXiv*, abs/1709.03968, 2018.

[6] Nabiha Asghar, Pascal Poupart, Jesse Hoey, Xin Jiang, and Lili Mou. Affective neural response generation. In *European Conference on Information Retrieval*, pages 154–166. Springer, 2018.

[7] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*, 2016.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[9] JinYeong Bak and Alice H. Oh. Variational hierarchical user-based conversation model. In *EMNLP/IJCNLP*, 2019.

[10] Suman Banerjee and Mitesh M Khapra. Graph convolutional network with sequential attention for goal-oriented dialogue systems. *Transactions of the Association for Computational Linguistics*, 7:485–500, 2019.

[11] R. Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34:1–34, 2008.

[12] Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[13] T. Bickmore and Rosalind W. Picard. Establishing and maintaining long-term human-computer relationships. *ACM Trans. Comput. Hum. Interact.*, 12:293–327, 2005.

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

[15] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[16] Antoine Bordes, Y-Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*, 2016.

[17] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[18] Antoine Bordes and Jason Weston. Learning end-to-end goal-oriented dialog. *CoRR*, abs/1605.07683, 2016.

[19] Antoine Bordes and Jason Weston. Learning end-to-end goal-oriented dialog. In *ICLR*, 2017.

[20] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[21] Petter Bae Brandtzæg and A. Følstad. Why people use chatbots. In *INSCI*, 2017.

[22] Paweł Budzianowski and Ivan Vulic. Hello, it's gpt-2 - how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. In *NGT@EMNLP-IJCNLP*, 2019.

[23] Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi. Skeleton-to-response: Dialogue generation guided by retrieval memory. *arXiv preprint arXiv:1809.05296*, 2018.

[24] Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. Retrieval-guided dialogue response generation via a matching-to-generation framework. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1866–1875, 2019.

[25] Yang Cai. Empathic computing. In *Ambient Intelligence in Everyday Life*, pages 67–85. Springer, 2006.

[26] Arun Chaganty, Stephen Mussmann, and Percy Liang. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, 2018.

[27] Zhangming Chan, Juntao Li, Xiaopeng Yang, Xiuying Chen, Wenpeng Hu, Dongyan Zhao, and Rui Yan. Modeling personalization in continuous space for response generation via augmented Wasserstein autoencoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1931–1940, Hong Kong, China, November 2019. Association for Computational Linguistics.

[28] Chaotao Chen, Jinhua Peng, Fan Wang, Jun Xu, and Hua Wu. Generating multiple diverse responses with multi-mapping and posterior mapping selection. *arXiv preprint arXiv:1906.01781*, 2019.

[29] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35, 2017.

[30] Hongshen Chen, Z. Ren, Jiliang Tang, Y. Zhao, and D. Yin. Hierarchical variational memory network for dialogue generation. *Proceedings of the 2018 World Wide Web Conference*, 2018.

[31] Hongshen Chen, Zhaochun Ren, Jiliang Tang, Yihong Eric Zhao, and Dawei Yin. Hierarchical variational memory network for dialogue generation. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 1653–1662, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[32] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[33] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[34] Kenneth Mark Colby. *Artificial paranoia: a computer simulation of paranoid process*. Pergamon Press, 1975.

[35] Kenneth Mark Colby, Sylvia Weber, and Franklin Dennis Hilf. Artificial paranoia. *Artif. Intell.*, 2:1–25, 1971.

[36] Pierre Colombo, Wojciech Witon, Ashutosh Modi, James Kennedy, and Mubbasir Kapadia. Affect-driven dialog generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3734–3743, 2019.

[37] Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. Question answering on knowledge bases and text using universal schema and memory networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–365, 2017.

[38] Nicola De Cao, Wilker Aziz, and Ivan Titov. Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2306–2317, 2019.

[39] Laurence Devillers, Ioana Vasilescu, and Lori Lamel. Annotation and detection of emotion in a task-oriented human-human dialog corpus. In *proceedings of ISLE Workshop*, 2002.

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[41] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. End-to-end reinforcement learning of dialogue agents for information access. In *ACL*, 2017.

[42] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*, 2018.

[43] Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander H. Miller, Arthur Szlam, and Jason Weston. Evaluating prerequisite qualities for learning end-to-end dialog systems. *CoRR*, abs/1511.06931, 2015.

[44] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*.

[45] Nouha Dziri, Ehsan Kamalloo, K. Mathewson, and Osmar R Zaiane. Evaluating coherence in dialogue systems using entailment. In *NAACL-HLT*, 2019.

[46] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[47] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, 2017.

[48] K. K. Fitzpatrick, Alison M Darcy, and Molly Vierhile. Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): A randomized controlled trial. *JMIR Mental Health*, 4, 2017.

[49] Pascale Fung, Dario Bertero, Yan Wan, Anik Dey, Ricky Ho Yin Chan, Farhad Bin Siddique, Yang Yang, Chien-Sheng Wu, and Ruixi Lin. Towards empathetic human-robot interactions. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 173–193. Springer, 2016.

[50] Sudeep Gandhe and D. Traum. Evaluation understudy for dialogue coherence models. In *SIGDIAL Workshop*, 2008.

[51] Jianfeng Gao, Michel Galley, and Lihong Li. Neural approaches to conversational ai. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 2–7, 2018.

[52] Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li, Guodong Zhou, and Shuming Shi. A discrete cvae for response generation on short-text conversation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1898–1908, 2019.

[53] Matt Gardner and Jayant Krishnamurthy. Open-vocabulary semantic parsing with both distributional statistics and formal knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3195–3201, 2017.

[54] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. A knowledge-grounded neural conversation model. *arXiv preprint arXiv:1702.01932*, 2017.

[55] Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. Dialoguegcn: A graph convolutional neural network for emotion recognition in conversation. *arXiv preprint arXiv:1908.11540*, 2019.

[56] Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. Affect-lm: A neural language model for customizable affective text generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 634–642, 2017.

[57] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[58] John J Godfrey, Edward C Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 517–520. IEEE, 1992.

[59] Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352. IEEE, 1996.

[60] Sergey Golovanov, R. Kurbanov, S. Nikolenko, Kyryl Truskovskyi, Alexander Tselousov, and T. Wolf. Large-scale transfer learning for natural language generation. In *ACL*, 2019.

[61] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *CoRR*, abs/1406.2661, 2014.

[62] Jia-Chen Gu, Tianda Li, Quan Liu, Xiaodan Zhu, Zhen-Hua Ling, Zhiming Su, and Si Wei. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. *arXiv preprint arXiv:2004.03588*, 2020.

[63] Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. Interactive matching network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the*

28th ACM International Conference on Information and Knowledge Management, pages 2321–2324, 2019.

[64] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1631–1640, 2016.

[65] Jing Gu, Qingyang Wu, and Zhou Yu. Perception score, a learned metric for open-ended text generation evaluation. *arXiv preprint arXiv:2008.03082*, 2020.

[66] Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. *arXiv preprint arXiv:1805.12352*, 2018.

[67] Fenfei Guo, Angeliki Metallinou, Chandra Khatri, Anirudh Raju, Anu Venkatesh, and Ashwin Ram. Topic-based evaluation for conversational bots. *arXiv preprint arXiv:1801.03622*, 2018.

[68] He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *ACL*, 2017.

[69] He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *ACL*, 2017.

[70] Behnam Hedayatnia, Seokhwan Kim, Yang Liu, Karthik Gopalakrishnan, Mihail Eric, and Dilek Hakkani-Tur. Policy-driven neural response generation for knowledge-grounded dialogue systems. *arXiv preprint arXiv:2005.12529*, 2020.

[71] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. Tracking the world state with recurrent entity networks. In *ICLR*, 2017.

[72] Matthew Henderson, Blaise Thomson, and Steve Young. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 360–365. IEEE, 2014.

[73] Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. Training neural response selection for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5392–5404, 2019.

[74] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[75] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050, 2014.

[76] Tianran Hu, A. Xu, Z. Liu, Quanzeng You, Yufan Guo, V. Sinha, Jiebo Luo, and R. Akkiraju. Touch your heart: A tone-aware chatbot for customer care on social media. *ArXiv*, abs/1803.02952, 2018.

[77] Wenpeng Hu, Zhangming Chan, Bing Liu, Dongyan Zhao, Jinwen Ma, and Rui Yan. Gsn: A graph-structured network for multi-party dialogues. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pages 5010–5016. AAAI Press, 2019.

[78] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric Xing. Toward controlled generation of text. In *ICML*, 2017.

[79] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596, 2017.

[80] Binxuan Huang and Kathleen M Carley. Syntax-aware aspect level sentiment classification with graph attention networks. *arXiv preprint arXiv:1909.02606*, 2019.

[81] Lishan Huang, Zheng Ye, Jinghui Qin, Liang Lin, and Xiaodan Liang. Grade: Automatic graph-enhanced coherence metric for evaluating open-domain dialogue systems. In *EMNLP*, 2020.

[82] Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32, 2020.

[83] B. Huber, D. McDuff, Chris Brockett, Michel Galley, and W. Dolan. Emotional dialogue generation using image-grounded language models. In *CHI '18*, 2018.

[84] Sina Jafarpour, Christopher JC Burges, and Alan Ritter. Filter, rank, and transfer the knowledge: Learning to chat. *Advances in Ranking*, 10, 2010.

[85] Mohit Jain, Pratyush Kumar, R. Kota, and S. Patel. Evaluating and informing the design of chatbots. *Proceedings of the 2018 Designing Interactive Systems Conference*, 2018.

[86] Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A. Smith. Dynamic entity representations in neural language models. In *EMNLP*, 2017.

[87] Zongcheng Ji, Zhengdong Lu, and Hang Li. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*, 2014.

[88] Jörg Tiedemann. *News from OPUS — A Collection of Multilingual Parallel Corpora with Tools and Interfaces*. 2009.

[89] Anjuli Kannan and Oriol Vinyals. Adversarial evaluation of dialogue models. In *Workshop on Adversarial Learning. NIPS*, 2016.

[90] Corey Lee M Keyes. Social well-being. *Social psychology quarterly*, pages 121–140, 1998.

[91] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations*, 2014.

[92] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations*, 2014.

[93] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[94] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[95] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[96] Jonathan Klein, Y. Moon, and Rosalind W. Picard. This computer responds to user frustration: Theory, design, and results. *Interact. Comput.*, 14:119–140, 2001.

[97] Sosuke Kobayashi, Ran Tian, Naoaki Okazaki, and Kentaro Inui. Dynamic entity representation with max-pooling improves machine reading. In *HLT-NAACL*, 2016.

[98] Stasinos Konstantopoulos. An embodied dialogue system with personality and emotions. In *Proceedings of the 2010 Workshop on Companionable Dialogue Systems*, pages 31–36. Association for Computational Linguistics, 2010.

[99] Sungjin Lee. Structured discriminative model for dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 442–451, 2013.

[100] Jia Li, Xiao Sun, Xing Wei, Changliang Li, and Jianhua Tao. Reinforcement learning based emotional editing constraint conversation generation. *arXiv preprint arXiv:1904.08061*, 2019.

[101] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of NAACL-HLT*, pages 110–119, 2016.

[102] Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. A persona-based neural conversation model. In *ACL*, 2016.

[103] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1106–1115, 2015.

[104] Jiwei Li, Will Monroe, and Dan Jurafsky. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*, 2016.

[105] Jiwei Li, Will Monroe, Alan Ritter, Daniel Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In *EMNLP*, 2016.

[106] Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

[107] Lihong Li, Jason D Williams, and Suhrid Balakrishnan. Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[108] Piji Li. An empirical investigation of pre-trained transformer language models for open-domain dialogue generation. *arXiv preprint arXiv:2003.04195*, 2020.

[109] Yan-Ran Li, Ruixiang Zhang, Wen-Jie Li, and Ziqiang Cao. Hierarchical prediction and adversarial learning for conditional response generation. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.

[110] Yanran Li and Wenjie Li. Meta-path augmented response generation. In *AAAI*, 2019.

[111] Yanran Li, Wenjie Li, Ziqiang Cao, and Chengyao Chen. Incorporating relevant knowledge in context modeling and response generation. *arXiv preprint arXiv:1811.03729*, 2018.

[112] Yanran Li, Wenjie Li, and Zhitao Wang. Graph-structured context understanding for knowledge-grounded response generation. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[113] Yanran Li, Hui Su, Xiaoyu Shen, and Wenjie Li. Dailydialog: A manually labelled multi-turn dialogue dataset. In *IJCNLP*, 2017.

[114] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*, 2004.

[115] Yankai Lin, Zhiyuan Liu, Huan-Bo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. In *EMNLP*, 2015.

[116] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 2015.

[117] Zhaojiang Lin, Peng Xu, Genta Indra Winata, Zihan Liu, and Pascale Fung. Caire: An end-to-end empathetic chatbot. *arXiv preprint arXiv:1907.12108*, 2019.

[118] Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2060–2069, 2018.

[119] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, 2016.

[120] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*, 2016.

[121] Pengfei Liu, Shuaichen Chang, Xuanjing Huang, Jian Tang, and Jackie Chi Kit Cheung. Contextualized non-local neural networks for sequence learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6762–6769, 2019.

[122] Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. Knowledge diffusion for neural dialogue generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1498, 2018.

[123] Yahui Liu, Wei Bi, Jun Gao, Xiaojiang Liu, Jian Yao, and Shuming Shi. Towards less generic responses in neural conversation models: A statistical reweighting method. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2769–2774, 2018.

[124] Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*, 2016.

[125] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[126] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *SIGDIAL Conference*, 2015.

[127] Ryan Lowe, Nissan Pow, Iulian V Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 285, 2015.

[128] Yi Luan, Yangfeng Ji, and Mari Ostendorf. Lstm based conversation models. *CoRR*, abs/1603.09457, 2016.

[129] E. Luger and A. Sellen. "like having a really bad pa": The gulf between user expectation and experience of conversational agents. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016.

[130] Chuwei Luo and Wenjie Li. A combination of similarity and rule-based method of polyu for ntcir-12 stc task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies*, 2016.

[131] Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

[132] Alex Luu and Sophia A. Malamud. Non-topical coherence in social talk: A call for dialogue model enrichment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 118–133, Online, July 2020. Association for Computational Linguistics.

[133] Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th Annual*

Meeting of the Association for Computational Linguistics, pages 5454–5459, Florence, Italy, July 2019. Association for Computational Linguistics.

[134] Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *ACL*, 2018.

[135] Christopher D. Manning and Mihail Eric. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. In *EACL*, 2017.

[136] Diego Marcheggiani, Joost Bastings, and Ivan Titov. Exploiting semantics in neural machine translation with graph convolutional networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 486–492, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[137] Abraham Harold Maslow. *A theory of human motivation*. Simon and Schuster, 2013.

[138] John D Mayer and Peter Salovey. The intelligence of emotional intelligence. *intelligence*, 17(4):433–442, 1993.

[139] John D Mayer, Peter Salovey, Susan Gomberg-Kaufman, and Kathleen Blainey. A broader conception of mood experience. *Journal of personality and social psychology*, 60(1):100, 1991.

[140] Indrani Medhi-Thies, N. Menon, S. Magapu, Manisha Subramony, and J. O'Neill. How do you want your chatbot? an exploratory wizard-of-oz study with young, urban indians. In *INTERACT*, 2017.

[141] Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. Coherent dialogue with attention-based language models. In *AAAI*, 2017.

[142] Mohsen Mesgar, Edwin Simpson, Yue Wang, and Iryna Gurevych. Generating persona-consistent dialogue responses using deep reinforcement learning. *arXiv preprint arXiv:2005.00036*, 2020.

[143] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539, 2015.

[144] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[145] Kaixiang Mo, Shuangyin Li, Yu Zhang, Jiajun Li, and Qiang Yang. Personalizing a dialogue system with transfer learning. *CoRR*, abs/1610.02891, 2016.

[146] Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M Khapra. Towards exploiting background knowledge for building conversation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332, 2018.

[147] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854, 2019.

[148] M. W. Morris and D. Keltner. How emotions work: The social functions of emotional expression in negotiations. *Research in Organizational Behavior*, 22:1–50, 2000.

[149] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136, Berlin, Germany, August 2016. Association for Computational Linguistics.

[150] Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. *arXiv preprint arXiv:1607.00970*, 2016.

[151] A. Muresan and H. Pohl. Chats with bots: Balancing imitation and engagement. *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

[152] Kevin Patrick Murphy and Stuart Russell. Dynamic bayesian networks: representation, inference and learning. 2002.

[153] Ryo Nakamura, Katsuhito Sudoh, Koichiro Yoshino, and Satoshi Nakamura. Another diversity-promoting objective function for neural dialogue generation. *arXiv preprint arXiv:1811.08100*, 2018.

[154] Mario Neururer, S. Schlögl, Luisa Brinkschulte, and Aleksander Groth. Perceptions on authenticity in chat bots. 2018.

[155] Xing Niu, Xinruo Sun, Haofen Wang, Shu Rong, Guilin Qi, and Yong Yu. Zhishi.me - weaving chinese linking open data. In *International Semantic Web Conference*, 2011.

[156] Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for nlg. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, 2017.

[157] Alice H Oh and Alexander I Rudnicky. Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*, pages 27–32. Association for Computational Linguistics, 2000.

[158] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):694–707, 2016.

[159] Gaurav Pandey, Dinesh Raghu, and S. Joshi. Mask focus: Conversation modelling by learning concepts. *ArXiv*, abs/2003.04976, 2020.

[160] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2793–2799, 2016.

[161] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318. Association for Computational Linguistics, 2002.

[162] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318. Association for Computational Linguistics, 2002.

[163] Timo Partala and Veikko Surakka. The effects of affective interventions in human–computer interaction. *Interacting with computers*, 16(2):295–309, 2004.

[164] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[165] Jiaxin Pei and Chenliang Li. S2spmn: a simple and effective framework for response generation with relevant information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 745–750, 2018.

[166] Shuang Peng, Hengbin Cui, Niantao Xie, Sujian Li, Jiaxing Zhang, and Xiaolong Li. Enhanced-rcnn: An efficient method for learning sentence similarity. In *Proceedings of The Web Conference 2020*, pages 2500–2506, 2020.

[167] Diana Perez-Marin and Ismael Pascual-Nieto. Conversational agents and natural language interaction: Techniques and effective. 2011.

[168] Volha Petukhova, Martin Gropp, Dietrich Klakow, Anna Schmidt, Gregor Eigner, Mario Topf, Stefan Srb, Petr Motlicek, Blaise Potard, John Dines, et al. The dbox corpus collection of spoken human-human and human-machine dialogues. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC" 14)*, number EPFL-CONF-201766. European Language Resources Association (ELRA), 2014.

[169] Rosalind W. Picard. Affective computing. 1997.

[170] Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, 7:100943–100953, 2019.

[171] Qiao Qian, Minlie Huang, and Xiaoyan Zhu. Assigning personality/identity to a chatting machine for coherent conversation generation. *arXiv preprint arXiv:1706.02861*, 2017.

[172] Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, Bill Dolan, Yejin Choi, and Jianfeng Gao. Conversing by reading: Contentful neural conversation with on-demand machine reading. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5427–5436, 2019.

[173] Libo Qin, Yijia Liu, W. Che, Haoyang Wen, Yangming Li, and Ting Liu. Entity-consistent end-to-end task-oriented dialogue system with kb retriever. In *EMNLP/IJCNLP*, 2019.

[174] Lisong Qiu, Juntao Li, Wei Bi, Dongyan Zhao, and Rui Yan. Are training samples correlated? learning to generate dialogue responses with multiple references. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 3826–3835, 2019.

[175] Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. Alime chat: A sequence to sequence and rerank based chatbot engine. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 498–503, 2017.

[176] Xipeng Qiu and Xuanjing Huang. Convolutional neural tensor network architecture for community-based question answering. In *Twenty-Fourth international joint conference on artificial intelligence*, 2015.

[177] A. Radford. Improving language understanding by generative pre-training. 2018.

[178] Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*, 2018.

[179] Christian Raymond and Giuseppe Riccardi. Generative and discriminative algorithms for spoken language understanding. In *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[180] Revanth Reddy, Danish Contractor, Dinesh Raghu, and Sachindra Joshi. Multi-level memory for task oriented dialogs. In *NAACL-HLT*, 2019.

[181] Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

[182] Danilo Jimenez Rezende, S. Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[183] Eric K Ringger, James F Allen, Bradford W Miller, and Teresa Sikorski. A robust system for natural spoken dialogue. 1996.

[184] Alan Ritter, Colin Cherry, and William B Dolan. Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*, pages 583–593. Association for Computational Linguistics, 2011.

[185] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020.

[186] Pum-Mo Ryu, Myung-Gil Jang, and Hyun-Ki Kim. Open domain question answering using wikipedia-based knowledge model. *Information Processing & Management*, 50(5):683–692, 2014.

[187] H. Sacks, E. Schegloff, and G. Jefferson. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50:696–735, 1974.

[188] Iulian Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron C. Courville. Multiresolution recurrent neural networks: An application to dialogue response generation. In *AAAI*, 2017.

[189] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. *arXiv preprint arXiv:1605.06069*, 2016.

[190] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1577–1586, 2015.

[191] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. In *ACL*, 2015.

[192] Roman Shantala, Gennadiv Kyselov, and Anna Kyselova. Neural dialogue system with emotion embeddings. In *2018 IEEE First International Conference on System Analysis & Intelligent Computing (SAIC)*, pages 1–4. IEEE, 2018.

[193] Fumin Shen, Wei Liu, Shaoting Zhang, Yang Yang, and Heng Tao Shen. Learning binary codes for maximum inner product search. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4148–4156, 2015.

[194] Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 2015.

[195] Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. A conditional variational framework for dialog generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 504–509, 2017.

[196] Stuart M. Shieber. Lessons from a restricted turing test. *Commun. ACM*, 37:70–78, 1994.

[197] Tetsuo Shinozaki, Yukiko Yamamoto, Setsuo Tsuruta, and Ernesto Damiani. An emotional word focused counseling agent and its evaluation. In *SMC*, 2014.

[198] Anshumali Shrivastava and Ping Li. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In *Advances in neural information processing systems*, pages 2321–2329, 2014.

[199] Heung-Yeung Shum, Xiao-dong He, and Di Li. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26, 2018.

[200] Heung-Yeung Shum, Xiaodong He, and Di Li. From eliza to xiaoice: Challenges and opportunities with social chatbots. *arXiv preprint arXiv:1801.01957*, 2018.

[201] Eric Michael Smith, Mary F. Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *ACL*, 2020.

[202] David R So, Chen Liang, and Quoc V Le. The evolved transformer. *arXiv preprint arXiv:1901.11117*, 2019.

[203] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

[204] Yiping Song, Zequn Liu, Wei Bi, Rui Yan, and Ming Zhang. Learning to customize language model for generation-based dialog systems. *arXiv preprint arXiv:1910.14326*, 2019.

[205] Yiping Song, Rui Yan, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, and Dongyan Zhao. An ensemble of retrieval-based and generation-based human-computer conversation systems. 2018.

[206] Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and X. Huang. Generating responses with a specific emotion in dialog. In *ACL*, 2019.

[207] Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and Xuan-Jing Huang. Generating responses with a specific emotion in dialog. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3695, 2019.

[208] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Cikm'15 Proceedings of the 24th Acm International on Conference on Information and Knowledge Management*. Association for Computing Machinery, 2015.

[209] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and William B. Dolan. A neural network approach to context-sensitive generation of conversational responses. In *HLT-NAACL*, 2015.

[210] Pei-Hao Su, Milica Gasic, Nikola Mrkšić, Lina M 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 *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2431–2441, 2016.

[211] Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. Continuously learning neural dialogue management. *arXiv preprint arXiv:1606.02689*, 2016.

[212] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

[213] Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, 2018.

[214] Xiao Sun, Xinmiao Chen, Zhengmeng Pei, and Fuji Ren. Emotional human machine conversation generation based on seqgan. In *2018 First Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia)*, pages 1–6. IEEE, 2018.

[215] Yizhou Sun, Rick Barber, Manish Gupta, Charu C Aggarwal, and Jiawei Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 121–128. IEEE, 2011.

[216] Yizhou Sun and Jiawei Han. Meta-path-based search and mining in heterogeneous information networks. *Tsinghua Science and Technology*, 18(4):329–338, 2013.

[217] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[218] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[219] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*, 2015.

[220] Jiliang Tang, Xia Hu, and Huan Liu. Social recommendation: a review. *Social Network Analysis and Mining*, 3(4):1113–1133, 2013.

[221] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019.

[222] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. One time of interaction may not be enough: Go deep with an interaction-over-interaction network for response selection in dialogues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1–11, 2019.

[223] Zhiliang Tian, Rui Yan, Lili Mou, Yiping Song, Yansong Feng, and Dongyan Zhao. How to make context more useful? an empirical study on context-aware neural conversational models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–236, 2017.

[224] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *ACL*, 2016.

[225] Yi-Lin Tuan, Yun-Nung Chen, and Hung-yi Lee. Dykgchat: Benchmarking dialogue generation grounding on dynamic knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1855–1865, 2019.

[226] Svitlana Vakulenko, M. Rijke, Michael Cochez, V. Savenkov, and A. Polleres. Measuring semantic coherence of a conversation. *ArXiv*, abs/1806.06411, 2018.

[227] Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3308–3318, 2019.

[228] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[229] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[230] Jeroen K Vermunt. Multilevel latent variable modeling: An application in education testing. *Austrian Journal of Statistics*, 37(3-4):285–299, 2008.

[231] Ashwin K. Vijayakumar, Michael Cogswell, R. R. Selvaraju, Q. Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *ArXiv*, abs/1610.02424, 2016.

[232] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700, 2015.

[233] Oriol Vinyals and Quoc V Le. A neural conversational model. In *Deep Learning Workshop*, 2015.

[234] Pavlos Vougiouklis, Jonathon S. Hare, and Elena Paslaru Bontas Simperl. A neural network approach for knowledge-driven response generation. In *COLING*, 2016.

[235] Marilyn A Walker, Owen C Rambow, and Monica Rogati. Training a sentence planner for spoken dialogue using boosting. *Computer Speech & Language*, 16(3-4):409–433, 2002.

[236] Richard S Wallace. The anatomy of alice. In *Parsing the Turing Test*, pages 181–210. Springer, 2009.

[237] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. Match-srnn: modeling the recursive matching structure with spatial rnn. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2922–2928, 2016.

[238] Di Wang and Eric Nyberg. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 707–712, 2015.

[239] Hai Wang, Takeshi Onishi, Kevin Gimpel, and David A. McAllester. Emergent predication structure in hidden state vectors of neural readers. In *Rep4NLP@ACL*, 2017.

[240] Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. A dataset for research on short-text conversations. In *EMNLP*, pages 935–945, 2013.

[241] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[242] Weichao Wang, Shi Feng, Daling Wang, and Yifei Zhang. Answer-guided and semantic coherent question generation in open-domain conversation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5066–5076, Hong Kong, China, November 2019. Association for Computational Linguistics.

[243] Weikang Wang, Jiajun Zhang, Qian Li, Chengqing Zong, and Zhifei Li. Are you for real? detecting identity fraud via dialogue interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1762–1771, 2019.

[244] Ye-Yi Wang and Alex Acero. Discriminative models for spoken language understanding. In *Ninth International Conference on Spoken Language Processing*, 2006.

[245] Yida Wang, Pei Ke, Yinhe Zheng, Kaili Huang, Y. Jiang, X. Zhu, and Minlie Huang. A large-scale chinese short-text conversation dataset. *ArXiv*, abs/2008.03946, 2020.

[246] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 2014.

[247] Joseph Weizenbaum. Eliza - a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9:36–45, 1966.

[248] Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, 2019.

[249] Haoyang Wen, Yijia Liu, Wanxiang Che, Libo Qin, and Ting Liu. Sequence-to-sequence learning for task-oriented dialogue with dialogue state representation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3781–3792, 2018.

[250] Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *arXiv preprint arXiv:1508.01755*, 2015.

[251] Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve Young. Latent intention dialogue models. In *ICML*, 2017.

[252] 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 *EACL*, pages 438–449, Valencia, Spain, April 2017. Association for Computational Linguistics.

[253] Jason D Williams. The best of both worlds: Unifying conventional dialog systems and pomdps. In *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[254] Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *arXiv preprint arXiv:1702.03274*, 2017.

[255] Jason D Williams and Geoffrey Zweig. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*, 2016.

[256] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[257] Chien-Sheng Wu, Richard Socher, and Caiming Xiong. Global-to-local memory pointer networks for task-oriented dialogue. *ArXiv*, abs/1901.04713, 2019.

[258] Sixing Wu, Ying Li, D. Zhang, Y. Zhou, and Zhonghai Wu. Topicka: Generating commonsense knowledge-aware dialogue responses towards the recommended topic fact. In *IJCAI*, 2020.

[259] Sixing Wu, Ying Li, Dawei Zhang, Yang Zhou, and Zhonghai Wu. Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In *ACL*, 2020.

[260] Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804, 2019.

[261] Xianchao Wu, Ander Martinez, and Momo Klyen. Dialog generation using multi-turn reasoning neural networks. In *HLT-NAACL*, 2018.

[262] Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou. Response generation by context-aware prototype editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7281–7288, 2019.

[263] Yu Wu, Wei Wu, Chen Xing, Can Xu, Zhoujun Li, and Ming Zhou. A sequential matching framework for multi-turn response selection in retrieval-based chatbots. *Computational Linguistics*, 45(1):163–197, 2019.

[264] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 496–505, 2017.

[265] Yu Wu, Wei Wu, Dejian Yang, Can Xu, Zhoujun Li, and Ming Zhou. Neural response generation with dynamic vocabularies. *arXiv preprint arXiv:1711.11191*, 2017.

[266] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[267] Y. Xie, Ekaterina Svikhnushina, and P. Pu. A multi-turn emotionally engaging dialog model. *ArXiv*, abs/1908.07816, 2019.

[268] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. Topic augmented neural response generation with a joint attention mechanism. *URL http://arxiv. org/abs/1606.08340*, 2016.

[269] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. Topic aware neural response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[270] Bo Xu, Yong Xu, Jiaqing Liang, Chenhao Xie, Bin Liang, Wanyun Cui, and Yanghua Xiao. Cn-dbpedia: A never-ending chinese knowledge extraction system. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 428–438. Springer, 2017.

[271] Can Xu, Wei Wu, Chongyang Tao, Huang Hu, Matt Schuerman, and Ying Wang. Neural response generation with meta-words. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5416–5426, 2019.

[272] Can Xu, Wei Wu, and Yu Wu. Towards explainable and controllable open domain dialogue generation with dialogue acts. *arXiv preprint arXiv:1807.07255*, 2018.

[273] J. Xu, Zeyang Lei, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and W. Che. Enhancing dialog coherence with event graph grounded content planning. In *IJCAI*, 2020.

[274] J. Xu, H. Wang, Zheng-Yu Niu, Hua Wu, and W. Che. Knowledge graph grounded goal planning for open-domain conversation generation. In *AAAI*, 2020.

[275] Puyang Xu and Ruhi Sarikaya. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 78–83. IEEE, 2013.

[276] Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie SUN SUN, Xiaolong Wang, Zhuoran Wang, and Chao Qi. Neural response generation via gan with an approximate embedding layer. In *EMNLP*, 2017.

[277] Rui Yan. " chitty-chitty-chat bot": Deep learning for conversational ai. In *IJCAI*, volume 18, pages 5520–5526, 2018.

[278] Rui Yan, Yiping Song, and Hua Wu. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 55–64. ACM, 2016.

[279] Rui Yan, Yiping Song, and Hua Wu. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *SIGIR*, 2016.

[280] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. *arXiv preprint arXiv:1512.00570*, 2015.

[281] Zhao Yan, Nan Duan, Junwei Bao, Peng Chen, Ming Zhou, Zhoujun Li, and Jianshe Zhou. Docchat: an information retrieval approach for chatbot engines using unstructured documents. In *ACL*, 2016.

[282] Bishan Yang and Tom Mitchell. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1436–1446, 2017.

[283] Bishan Yang and Tom Mitchell. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446, 2017.

[284] Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. Reference-aware language models. In *EMNLP*, 2016.

[285] Kaisheng Yao, Baolin Peng, Geoffrey Zweig, and Kam-Fai Wong. An attentional neural conversation model with improved specificity. *CoRR*, abs/1606.01292, 2016.

[286] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. Recurrent neural networks for language understanding. In *Interspeech*, pages 2524–2528, 2013.

[287] Lili Yao, Yaoyuan Zhang, Yansong Feng, Dongyan Zhao, and Rui Yan. Towards implicit content-introducing for generative short-text conversation systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2190–2199, 2017.

[288] Lili Yao, Yaoyuan Zhang, Yansong Feng, Dongyan Zhao, and Rui Yan. Towards implicit content-introducing for generative short-text conversation systems. In *EMNLP*, 2017.

[289] Semih Yavuz, Abhinav Rastogi, Guan-Lin Chao, and Dilek Hakkani-Tur. Deepcopy: Grounded response generation with hierarchical pointer networks. *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, 2019.

[290] Tom Young, Erik Cambria, Iti Chaturvedi, Minlie Huang, Hao Zhou, and Subham Biswas. Augmenting end-to-end dialog systems with commonsense knowledge. *arXiv preprint arXiv:1709.05453*, 2017.

[291] Dian Yu, Michelle Cohn, Yi Mang Yang, Chun Yen Chen, Weiming Wen, Jiaping Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhowmick, et al. Gunrock: A social bot for complex and engaging long conversations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 79–84, 2019.

[292] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.

[293] Xiao Yu, Xiang Ren, Quanquan Gu, Yizhou Sun, and Jiawei Han. Collaborative filtering with entity similarity regularization in heterogeneous information networks. *IJCAI HINA*, 27, 2013.

[294] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 283–292, 2014.

[295] Zhou Yu, Ziyu Xu, Alan W Black, and Alexander Rudnicky. Strategy and policy learning for non-task-oriented conversational systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 404–412, 2016.

[296] Jennifer Zamora. I'm sorry, dave, i'm afraid i can't do that: Chatbot perception and expectations. *Proceedings of the 5th International Conference on Human Agent Interaction*, 2017.

[297] Hainan Zhang, Yanyan Lan, J. Guo, J. Xu, and X. Cheng. Reinforcing coherence for sequence to sequence model in dialogue generation. In *IJCAI*, 2018.

[298] Jiangtao Zhang, Juanzi Li, Xiao-Li Li, Yao Shi, Junpeng Li, and Zhigang Wang. Domain-specific entity linking via fake named entity detection. In *International Conference on Database Systems for Advanced Applications*, pages 101–116. Springer, 2016.

[299] Jiayi Zhang, Chongyang Tao, Zhenjing Xu, Qiaojing Xie, Wei Chen, and Rui Yan. Ensemblegan: Adversarial learning for retrieval-generation ensemble model on short-text conversation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 435–444, 2019.

[300] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[301] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[302] Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems*, pages 1810–1820, 2018.

[303] Yizhe Zhang, S. Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and W. Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. *ArXiv*, abs/1911.00536, 2020.

[304] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*, 2019.

[305] Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3740–3752, 2018.

[306] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, 2017.

[307] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*, 2017.

[308] Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, 2019.

[309] Peixiang Zhong, D. Wang, and C. Miao. An affect-rich neural conversational model with biased attention and weighted cross-entropy loss. In *AAAI*, 2019.

[310] Ganbin Zhou, Ping Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. Mechanism-aware neural machine for dialogue response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[311] Ganbin Zhou, Ping Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. Mechanism-aware neural machine for dialogue response generation. In *AAAI*, 2017.

[312] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. Emotional chatting machine: Emotional conversation generation with internal and external memory. *arXiv preprint arXiv:1704.01074*, 2017.

[313] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629, 2018.

[314] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

[315] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93, 2020.

[316] Xianda Zhou and William Yang Wang. Mojitalk: Generating emotional responses at scale. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1128–1137, 2018.

[317] Xianda Zhou and William Yang Wang. Mojitalk: Generating emotional responses at scale. *ArXiv*, abs/1711.04090, 2018.

[318] Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. Multi-view response selection for human-computer conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 372–381, 2016.

[319] Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1127, 2018.

[320] Qingfu Zhu, Lei Cui, Weinan Zhang, Furu Wei, and Ting Liu. Retrieval-enhanced adversarial training for neural response generation. *arXiv preprint arXiv:1809.04276*, 2018.

[321] Qingfu Zhu, Lei Cui, Weinan Zhang, Furu Wei, and Ting Liu. Retrieval-enhanced adversarial training for neural response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3763–3773, 2019.

[322] Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. Flexible end-to-end dialogue system for knowledge grounded conversation. *arXiv preprint arXiv:1709.04264*, 2017.