University of Nottingham
UK | CHINA | MALAYSIA

Modeling Context and Knowledge for Dialogue Generation

Wen Zheng

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

Match 2023

To my parents: thank you for your constant love, support, and encouragement. Love you both!

Abstract

Dialogue generation reflects the cutting-edge AI technology application, with several AI assistants already developed by prominent Information and Communications Technology (ICT) companies like Google Assistant, Apple Siri, and Microsoft Cortana. However, AI support for open dialogue presents many challenges and research continues to address them, including the ultimate challenge of passing the Turing test. One of the key issues in dialogue generation is the source of content that can be used to train generative models within dialogue systems. Over the past decade, the construction of dialogue-generation datasets (e.g., Wizard of Wikipedia, CMU-DoG, DailyDialog, etc.) and research on models for context-aware and knowledge-based dialogue generation have been actively pursued by researchers. A detailed review of the related literature has revealed that the field has rapidly evolved, leading to significant progress but also giving rise to new questions and challenges. In this thesis, this line of research is continued, and specific issues in context-aware and knowledge-based dialogue generation are addressed, including: (1) Context Usage in Dialogue Generation, incorporating dialogue intrinsic properties related to the speakers and content characteristics; (2) Knowledge Injection in Dialogue Generation, enabling the incorporation of multiple sources of knowledge; (3) Knowledge Selection in Dialogue Generation, with flexibility to separate the injection of knowledge from dialogue generation; (4) Term-Level Knowledge De-noising in Dialogue Generation, simulating response representations that can be used for knowledge de-noising in the test phase; and (5) Differentiating Context Use for Knowledge Selection and Response Generation, supporting a distinct use of context and contextualized knowledge for selecting knowledge and generating responses.

This research resulted in new methods and models that represent key novel contributions of this thesis. (1) Starting from context-only dialogue generation, a context-aware dialogue generation model named GMATs was implemented, leveraging the dialogue's intrinsic characteristics such as speaker roles and part-of-speech indicators for the dialogue utterances. (2) Given the evidence that knowledge can help improve dialogue generation, a Transformer-based knowledge injection model, TED, was designed, featuring weights for different knowledge units. The conclusion that knowledge should not be injected indiscriminately but should be carefully selected can be obtained. (3) To address this issue, a knowledge selection mechanism, named TPPA, was explored, approximating a post-retrieved knowledge network with a response-retrieved knowledge network, enabling TPPA to emulate the ground-truth response and retrieve the relevant knowledge sentences. (4) Furthermore, an investigation into how to de-noise the injected knowledge at the term level was conducted, and a KTWM model was introduced to filter out noise during model training. (5) In the end, to construct a unified dialogue generation framework, the CKL model was proposed, built on the premise that context plays a role in the knowledge selection task that is different from its role in the dialogue generation task.

The effectiveness of various models for incorporating context and knowledge into dialogue generation was empirically investigated in this thesis. Specifically, the impact of context awareness, knowledge weighting, knowledge selection, term-level de-noising, and a unified model approach on the performance of the dialogue generation methods proposed in this thesis were evaluated. The experimental results demonstrate that the use of dialogue context is critical for improving the performance of response generation: A 12.8% improvement (over the Syntax-infused BART) in the BLEU-2 score was achieved by the GMATs model, considering dialogue intrinsic characteristics such as speaker role and part-of-speech. Additionally, compared to previous works, further performance improvements of 23.1% (TED vs. WSeq-Sum) and 4.4% (TPPA vs. TED) on the BLEU-2 and Meteor metrics, respectively, were achieved by assigning different weights to knowledge sentences (TED model) and selecting knowledge using the TPPA method. The proposed KTWM model for term-level de-noising also resulted in a 6.3% improvement in the BLEU-2 score on top of TED. Finally, the CKL model, which incorporates all of these approaches into a unified framework, outperformed the best previous study, DIALKI, by 15.2% on the BLEU-2 score and even surpassed TED by a large margin (86.5%). Overall, these findings suggest that differentiating context usage for the knowledge selection task and response generation task is critical when designing a dialogue generation model, and incorporating knowledge into dialogue systems using sentence-level knowledge selection and term-level de-noising can significantly enhance their performance.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Ke (Adam) Zhou, who has significantly impacted my scientific and working perspectives. Dr. Zhou has taught me the importance of taking the time to do things right in research, emphasizing the need for rigor and precision in every step of the research process. His guidance and support have been invaluable, not only in my research endeavors but also in my personal life, assisting me with scholarship applications, housing arrangements, and other aspects of my graduate journey.

I am deeply grateful to my supervisor, Prof. Natasa Milic-Frayling, who has devoted countless hours to reviewing and improving my papers. Prof. Milic-Frayling has been meticulous in her guidance, helping me understand the importance of conciseness and accuracy in my work and teaching me that not all experiments need to be successful; sometimes, failed experiments also contribute to the advancement of research by showing which directions are ineffective. In my heart, she is not only a supervisor but also one of my best friends. I wish her continued success in her career and, importantly, a happy life.

A big thank-you would be given to Yuan Tian, who has been of great help since before I began my Ph.D. program. Her extensive life experiences have been truly beneficial to me. To Zeyang Liu, my great roommate, thank him for the care when I was sick, the discussions during research challenges, and the companionship during our travels. I am grateful to Weiyao Meng for always sharing her warmth and positivity with everyone around her, of course including me. Thank you, Yuzheng Chen, for his impressive ability to remember different routes during our travels and his incredible patience. I also appreciate Feng Chen for showing us the various paths one can take in life and the unique experiences each can offer.

I want to thank Francis for introducing me to many historical sites throughout the United Kingdom, and patiently sharing the unique stories behind each tourist interest. I am grateful to Callum Plews and Ashley Dukart for their support during my low moments and the challenges of the Covid pandemic. I am also thankful for all the friends I have met at Grace Church, who have made my Ph.D. journey more colorful and fulfilling. To Wei Li, Jingzhi Wei, Linmao Liang, and Sitong Pan, I extend my big thank-you for your friendship since my arrival in the UK and our joyful year together in Nottingham.

I want to express my profound gratitude to my wife, Keqiao Wu, who is the most important person in my life moving forward. Her understanding of the difficulties encountered in research is deeply appreciated. During the numerous times when my experiments failed, I was always surrounded by her unwavering encouragement. I appreciate her steadfast support during our three years of a long-distance relationship, knowing full well the challenges of living alone as a single woman and facing the pressures of societal opinions. I am grateful for her willingness to start our life together from scratch as we embark on this new chapter after graduation, with her happiness and trust placed firmly in me. Carrying the responsibilities for our family and the love we share, I am confident that we will build a wonderful future together.

Finally, I want to extend my heartfelt thanks to my parents. It is because of their understanding, nurturing, and support that I have come this far. They have always told me, “Home is always your safe harbor” and this belief has given me the courage to move forward. Words cannot express the depth of gratitude and appreciation I have for my parents. I love you both!

List of Publications

1. Wen Zheng, Natasa Milic-Frayling, and Ke Zhou. 2023. GMATs Response Generation: Learning Dialogue Context Characteristics using Auxiliary Tasks (Submitted to EMNLP 2023) [Chapter 4]

2. Wen Zheng and Ke Zhou. Enhancing conversational dialogue models with grounded knowledge. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pages 709–718. [Chapter 5]

3. Wen Zheng, Natasa Milic-Frayling, and Ke Zhou. Approximation of response knowledge retrieval in knowledge-grounded dialogue generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pages 3581–3591. [Chapter 6]

4. Wen Zheng, Natasa Milic-Frayling, and Ke Zhou. Knowledge-grounded dialogue generation with term-level de-noising. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2972–2983. [Chapter 7]

5. Wen Zheng, Natasa Milic-Frayling, and Ke Zhou. Contextual Knowledge Learning for Dialogue Generation. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers): ACL 2023, pages 7822-7839. [Chapter 8]

Contents

Abstract **i**

Acknowledgements **iii**

List of Publications **v**

1 Introduction **1**

1.1 Dialogue Systems Types and Design Challenges 1

1.2 Research Questions 5

1.3 Contributions 11

1.4 Thesis Outline 13

2 Background **15**

2.1 Backbones of the Generative Models 15

2.1.1 Recurrent Neural Networks 15

2.1.2 Transformer 16

2.1.3 Pre-trained Models 17

2.2 Evaluation Metrics 18

2.2.1 Lexical Metrics 19

2.2.2 Embedding-based Metrics 21

2.2.3 Diversity Metric - Distinct Scores 22

2.3 Datasets 23

2.3.1 Wizard of Wikipedia 23

2.3.2 DailyDialog 24

2.3.3 Reddit 24

2.3.4 Holl-E 25

2.3.5 CMU-DoG 25

2.4 Key Notations 26

3 Related Work 27

3.1 Sequence-to-Sequence Models 27

3.2 Dialogue Characteristics 29

3.3 Context-Aware Dialogue Generation 29

3.4 Knowledge-Grounded Dialogue Generation 32

3.5 Knowledge Selection for Dialogue Generation 33

3.6 Large Language Models (LLMs) 34

3.7 Summary 35

4 Response Generation Method Given Context Information 37

4.1 Methodology 39

4.1.1 GMATs Encoder 39

4.1.2 GMATs Decoder 41

4.1.3 Auxiliary Tasks 43

4.1.4 Final Loss Function 45

4.2 Experiment 46

4.2.1 Datasets and Metrics 46

4.2.2 Implementation Details 46

4.2.3 Baseline Approaches 46

4.3 Experimental Results and Analysis 47

4.3.1 Dialogue Generation Results 47

4.3.2 Effectiveness of Each Component 49

4.4 Summary 50

5 Enhancing Conversational Dialogue Models with Grounded Knowledge 51

5.1 Fundamental Models 53

5.1.1 RNN-based Sequence-to-Sequence Models 53

5.1.2 End-to-End Memory Network 54

5.1.3 Transformer-based Models 56

5.1.4 Discussion on These Methods 57

5.2 Methodology . . . 58

5.3 Experiment . . . 60

5.3.1 Datasets . . . 60

5.3.2 Implementation Details . . . 60

5.3.3 Metrics . . . 62

5.3.4 Baseline Approaches . . . 63

5.4 Experimental Results and Analysis . . . 63

5.4.1 Results and Analysis of the Models . . . 64

5.4.2 Results and Analysis of The Right Amount of Knowledge Sentences . . . 66

5.4.3 The Impact of Noisy External Knowledge . . . 67

5.5 Summary . . . 68

6 Knowledge Selection for Knowledge-grounded Dialogue Generation 70

6.1 Motivations . . . 72

6.1.1 RRK Assessment on WoW Training Data . . . 72

6.2 TPPA Method . . . 74

6.2.1 TPPA Training phase . . . 76

6.2.2 TPPA Test Phase . . . 78

6.3 Experiment . . . 79

6.3.1 Datasets . . . 79

6.3.2 Implementation Details . . . 79

6.3.3 Metrics . . . 80

6.3.4 Baselines . . . 80

6.4 Experimental Results and Analysis . . . 81

6.4.1 Knowledge Selection Evaluation . . . 81

6.4.2 Response Generation Evaluation . . . 83

6.4.3 Upper-bound Analysis . . . 85

6.4.4 Analysis of Added Useful Words . . . 85

6.5 Summary . . . 87

7 Knowledge-Grounded Dialogue Generation with Term-level De-noising 89

7.1 Methodology . . . 91

7.1.1 Knowledge Selection and Representation . . . 93

7.1.2 Knowledge Term Weighting . . . 94

7.1.3 KTWM Decoder . . . 96

7.2 Experiment . . . 98

7.2.1 Datasets . . . 98

7.2.2 Implementation Details . . . 98

7.2.3 Metrics . . . 99

7.2.4 Baselines. . . . 99

7.2.5 Experiment Design . . . 100

7.3 Experimental Results and Analysis . . . 100

7.3.1 Performance of Generating Response . . . 100

7.3.2 Results of Knowledge Term Weighting . . . 101

7.3.3 Analysis of Input Sequence Length . . . 102

7.3.4 Good cases and bad cases of Knowledge Term Weights and KTWM Generated Responses . . . 103

7.4 Summary . . . 105

8 A Unified Dialogue Generation Model with Contextual Knowledge Learning 108

8.1 Methodology . . . 111

8.1.1 Encoder . . . 113

8.1.2 Context Latent Weight Generator . . . 114

8.1.3 Knowledge Latent Weight Generator . . . 116

8.1.4 Decoder and Training . . . 118

8.2 Experiment . . . 119

8.2.1 Datasets . . . 119

8.2.2 Implementation Details . . . 119

8.2.3 Metrics . . . 120

8.2.4 Baselines 121

8.3 Experimental Results and Analysis 122

8.3.1 Dialogue Generation Results on Automatic Evaluation and Human Evaluation. 122

8.3.2 Ablation Study 124

8.3.3 Latent Weight Analysis. 125

8.3.4 Low-Resource Experiments 127

8.3.5 Effect of Top-N Retrieved Knowledge 128

8.3.6 Case Study 128

8.4 Summary 130

9 Conclusions 131

9.1 Main Conclusions 131

9.2 Thinking about Further Works 134

Bibliography 139

Appendices 150

A List of Abbreviations 150

List of Tables

1.1 Research questions (RQ) in context-aware and knowledge-grounded dialogue generation, formulated based on literature review and research explorations. 7

2.1 Statistics of each dataset used in this thesis. ‘k’ means thousand, and ‘M’ denotes million. 23

2.2 Summary of the key notations used in the thesis. 26

4.1 Automatic evaluation results on DailyDialog and Wizard of Wikipedia datasets. * means significant test value with $p < 0.05$, in comparison with the proposed GMATs . ‘w/o’ means without a certain loss function for the ablation study. All values are expressed as percentages (%). 48

5.1 The red-tagged vocabulary in targets (responses) appears in knowledge, but not in the post. Incorporating the appropriate vocabulary from the knowledge, the proposed knowledge-grounded model (TED) is able to generate superior responses than the model (Vanilla Transformer) that disregards knowledge. 52

5.2 For the Filtered Reddit and Wizard of Wikipedia data sets: Compare different types of knowledge-grounded models on automatic metrics. TED is used as the base model to do the significant test. * stands for significant test value $p < 0.05$. . . . 63

5.3 Injecting a different number of knowledge sentences to the TED on the Wizard of Wikipedia dataset. The TED gets a peak performance at 3, so the 3-knowledge-sentence model is taken as the base model to do the significant test. 'KS' means knowledge sentence. * stands for significant test value $p < 0.05$. 66

5.4 Injecting more external knowledge to the WSeq-Sum model on the Wizard of Wikipedia dataset. The WSeq-Sum model gets a peak performance at 12 KS, so the 12 KS model is taken as the base model to do the significant test. 'KS' means knowledge sentences. * stands for significant test value $p < 0.05$. 66

5.5 For the Original Reddit dataset: Compare different types of knowledge-grounded models on 3 typical metrics. TED is taken as the base model to do the significant test. * stands for significant test value $p < 0.05$. 67

6.1 Example of a post and a response from the Wizard of Wikipedia (WoW) data set with top 2 ranked outputs from TPPA, the post-retrieved knowledge PRK, and the response-retrieved knowledge RRK. Blue indicate words present in the WoW response and RRK but not in PRK. 71

6.2 Injection of various sources into the Transformer and TED using WoW data set. All the values are percentages reported by the performance metrics (%). 74

6.3 Retrieval precision on the WoW and Holl-E data sets. '*' means t-test $p < 0.05$ compared with the baseline $BM25$; '†' is the $p < 0.05$ compared with the best performing group. **Bold** indicates the best performance group when changing the number of negative samples. Underline indicates the best group among all methods. 82

6.4 Performance of generative models MemNet, WSeq, and TED with the best TPPA knowledge selection. Post-KS and SKT rely on their jointly trained models. BS refers to Bert-Score. . . 83

6.5 Knowledge-injection results on the Wizard of Wikipedia data set. The values are percentages (%). '*' means the t-test $p < 0.05$ compared with the BM25 algorithm. 'Top 1', 'Top 5', and 'Top 10' denotes injecting top 1 or 5, or 10 ranking knowledge. BS is Bert-Score. **Bold** indicates the best score apart from the rrk_i group. 84

6.6 Knowledge-injection results on the Holl-E data set. The values are percentages (%). '*' means the t-test $p < 0.05$ compared with the BM25 algorithm. 'Top 1', 'Top 5', and 'Top 10' denotes injecting top 1 or 5, or 10 ranking knowledge. BS is Bert-Score. **Bold** indicates the best score apart from the rrk_i group. 86

6.7 The useful word overlapping rate results of WoW and Holl-E data sets. All values are shown as percentages (%). 87

7.1 Example of a post, ground truth response, injected knowledge, and generated response by the Knowledge Term Weighting Model (KTWM). The term highlights indicate the predicted probability of a term being useful. 90

7.2 KTWM performance on the WoW seen and unseen test data and Holl-E dataset with different knowledge sources. Comparison with Post-KS, SKT, and TED models. '*' indicates statistical significance ($p < 0.05$). **Bold** indicates the best performance for a given metric. 'w' denotes 'with', i.e., injecting the knowledge source that is used in a specific baseline model. 101

7.3 Precision, Recall, and F-1 scores for the useful and noisy term predictions on the WoW seen test set. 102

7.4 Ablation study of the multi-component loss function on the WoW seen test set. w/o means 'without'. 103

7.5 Positive cases that the injected knowledge is properly weighted by the proposed KTWM model. The background color stands for the probability of being a useful term. 106

7.6 Negative cases that the injected knowledge is not properly weighted by the proposed KTWM model. The background color stands for the probability of being a useful term. 107

8.1 Automatic evaluation results on Wizard of Wikipedia (WoW) test seen/unseen and CMU-DoG datasets. * means significant test value with $p < 0.05$, compared to the CKL. Note that the results of ITDD and DRD are copied from the papers, so they do not have significant test results. 'w/o' means without a certain module for the ablation study. All values are expressed as percentages (%). 123

8.2 Human evaluation on Wizard of Wikipedia test seen and CMU-DoG datasets. The values except for Kappa are in percentage (%). 124

8.3 Spearman's Correlation between latent weights and the ground truths. * means significant test value with $p < 0.05$, in comparison with the proposed CKL. 'w/o' mean models without a certain module. 126

8.4 Wizard of Wikipedia test seen & unseen and CMU-DoG evaluation results on low-resource scenarios. * means significant test value with $p < 0.05$, in comparison with the full version of CKL. All values are expressed as percentages (%). 127

8.5 A good case and a bad case of the proposed CKL model. In the base case, 'A' and 'B' denote the two Amazon MTurk workers exchanging their opinions. 129

List of Figures

1.1 Automated generative approaches and retrieval-based approaches for Task-Oriented and Open-Domain dialogue systems. The Open-Domain generative approaches are my focus. 3

1.2 A three-turn dialogue example that consists of context, i.e., dialogue utterances, the corresponding knowledge sentences, and the target response. Note that the last context utterance is viewed as Post. 4

4.1 Overview of the proposed GMATs model. 39

4.2 Architecture of the proposed GMATs model. Speaker Role Embedding and Part-of-Speech Embedding are introduced to indicate the speakers' roles and part-of-speech. Four auxiliary tasks are designed to facilitate learning general and specific aspects of content and speakers' characteristics. 41

5.1 The TED model: a) a Transformer encoder (the same as the vanilla Transformer); b) an expanded decoder. The right-hand side sub-figure is the detailed inner modules. For simplification, some modules are omitted in the right-hand side sub-figure. In b), the 'Extra Info' stands for external knowledge, and the 'Calc' means the probabilities calculating layer. 58

6.1 Common word count distribution between each source and the target response on the WoW training set. The dashed lines are the average count of common words in each group (after removing stop words). 73

6.2 TPPA Architecture comprises (1) Post Processing Unit, (2) Response Processing Unit (right), and (3) Knowledge Selection Unit (middle). 75

6.3 Common word count distribution between each source and the target response on the Reddit training set. The dashed lines are the average count of common words in each group (after removing stop words). 88

7.1 Overview of the Knowledge Term Weighting Model. V stands for representations for post, knowledge, and response. SRV means simulated response vector, and A_{sk} and A_{rk} denote knowledge terms' weights matrices assigned with SRV and response, respectively. 91

7.2 Architecture of the Knowledge Term Weighting Model (KTWM) showing the operations in the training and test phase. $\otimes$ designates matrix multiplication; $\odot$ designates element-wise multiplication. 92

7.3 Knowledge Term Weighting Model Decoder. 96

7.4 Effects of the increased number of knowledge terms on the KTWM performance (WoW seen test set). 102

8.1 Example showing utterances of participants A and B, the scored knowledge, and the target response. The knowledge sentence (1) is deemed the best knowledge for response generation by the proposed CKL model. The best context segments for retrieving the best knowledge are colored Brown; the best context segments for generating response are colored Blue and the best knowledge segments for the response are Purple. 109

8.2 Overview of the Contextual Knowledge Learning (CKL) model. 112

8.3 Contextual Knowledge Learning (CKL) model architecture comprising the BART-based encoder, two generators for latent vectors training and latent weight generation, and decoder with context and knowledge scoring. Trainable parameters δ_i balance multi-losses. 113

8.4 P@N scores for WoW test seen original knowledge set and the re-ranked knowledge set by the proposed CKL. 126

8.5 BLEU-2 score (Y-axis) for Wizard of Wikipedia and CMU-DoG under different numbers of top N retrieved knowledge sentences being the ground truth. 128

Chapter 1

Introduction

1.1 Dialogue Systems Types and Design Challenges

Dialogue systems, also known as conversational agents, have gained significant attention in recent years due to their growing role in enhancing human-computer interaction. These systems are designed to understand and process natural language, providing relevant and coherent responses within the context of the conversation. The field of dialogue systems can be broadly classified into two categories: task-oriented dialogue approaches and open-domain dialogue approaches. Each of them exhibits specific characteristics and caters to different applications and user requirements.

Task-oriented dialogue approaches are designed to assist users in accomplishing specific goals. A commercially available agent Microsoft Cortana [1] can, for example, operate a user's computer given voice input from the user. Amazon Echo [2] enables the user to play music in a hand-free manner. These approaches are typically built around a predefined domain, making them highly effective at handling conversations that pertain to specific tasks. By utilizing techniques like intent recognition and slot filling, task-oriented models can

[1] https://www.microsoft.com/cortana

[2] https://en.wikipedia.org/wiki/Amazon_Echo

extract pertinent information from user inputs, enabling them to provide accurate and concise assistance. Due to their focused nature, these models excel in delivering efficient and goal-oriented interactions.

Open-domain dialogue approaches, on the other hand, aim to engage users in more free-flowing, casual, and diverse conversations. These approaches are not limited to a specific domain or task, making them capable of discussing a broad range of topics. Open-domain models typically rely on massive datasets and advanced machine learning techniques, such as deep learning and reinforcement learning, to generate contextually appropriate responses. While these models may not always provide precise information, their ability to converse naturally and maintain engaging discussions makes them particularly well-suited for social and entertainment applications. For example, Microsoft's XiaoIce[3] can engage in open-ended conversations on any subject brought up by users, while Apple's Siri[4] is capable of entertaining children through verbal interactions, even if the content of the conversation is simplistic or juvenile.

The main features of dialogue approaches can be attributed to their underlying mechanisms and objectives. For both task-oriented and open-domain tasks, some key features include:

a) Natural Language Representation: Almost all neural network-based models begin with the process of obtaining representations. With the advent of the Transformer (Vaswani et al., 2017), natural language inputs are embedded and enriched by the self-attention mechanism. It has been demonstrated that improved representations lead to better performance.

b) Context Preservation: Dialogue models are designed to keep track of the conversational context, ensuring that the responses are relevant to the ongoing discussion and account for the information already exchanged between the speakers.

c) Incorporation of Background Knowledge: This characteristic allows di-

[3] https://www.msxiaobing.com/

[4] https://www.apple.com/siri/

alogue systems to access external information. Task-oriented dialogue systems typically retrieve data from databases, while open-domain dialogue systems are required to respond informatively based on relevant knowledge resources.

d) Response Generation / Retrieval: Generating or retrieving suitable, coherent, and contextually accurate responses is essential for sustaining meaningful conversations. Both task-driven and open-domain models utilize various methods, such as rule-based, template-based, or neural network-based approaches, to accomplish this objective.

These four aspects collectively constitute the foundations of a dialogue system design. The strategies presented in this thesis cover these essential characteristics in relation to the dialogue system I focus on.

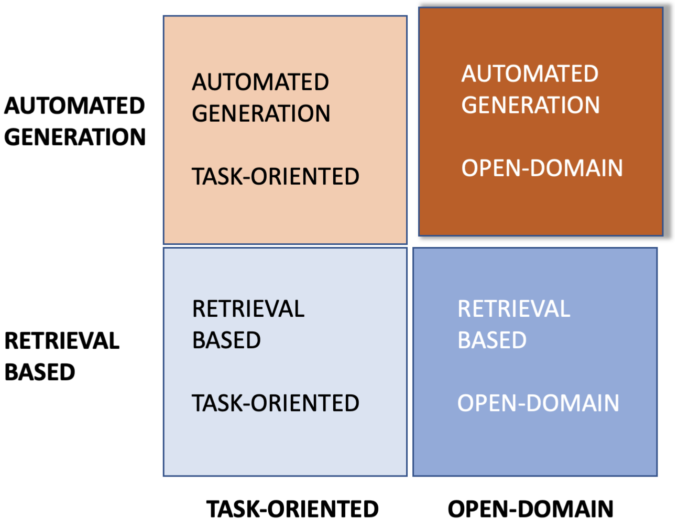
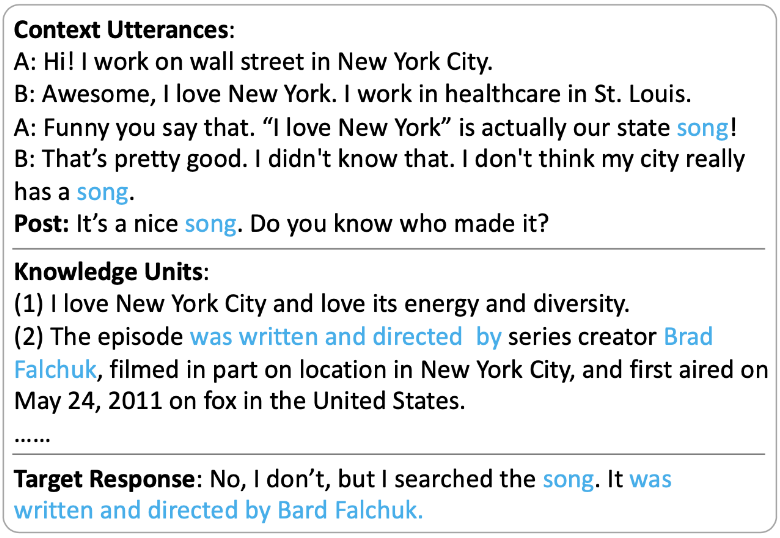

AUTOMATED GENERATION
RETRIEVAL BASED
AUTOMATED GENERATION TASK-ORIENTED
AUTOMATED GENERATION OPEN-DOMAIN
RETRIEVAL BASED TASK-ORIENTED
RETRIEVAL BASED OPEN-DOMAIN
TASK-ORIENTED
OPEN-DOMAIN

Figure 1.1: *Automated generative approaches and retrieval-based approaches for Task-Oriented and Open-Domain dialogue systems. The Open-Domain generative approaches are my focus.*

As highlighted in Figure 1.1, for both task-oriented dialogue approaches and open-domain dialogue approaches, retrieval-based methods and automated generation methods have been proposed. In this thesis, my primary focus is on the open-domain automated dialogue generation approaches. It typically involves the context of the dialogue, i.e., all the dialogue utterances, and the background knowledge that can be incorporated into the response generation process. This can be well reflected by the information-seeking scenarios in real life, e.g., search engine and customer services. When users want to search for information, the search engines are the first place to go. The inputted queries are casual and various across different users and also, their purposes are entirely different. The customer services have similar patterns. Both scenarios match the definition of open-domain conversation. Therefore, the research topics of the generative method matter a lot in our daily lives.

An example is illustrated in Figure 1.2, where two participants (speakers A and B) engage in a conversation with alternating statements. Speaker B responds to Speaker A while taking relevant knowledge units into account. In the target response, the blue keywords are present in both the context and knowledge. It is evident that the term 'song' in the response is mentioned in the context, while the phrase 'was written and directed by Bard Falchuk' is derived from the knowledge.



Context Utterances:
A: Hi! I work on wall street in New York City.
B: Awesome, I love New York. I work in healthcare in St. Louis.
A: Funny you say that. "I love New York" is actually our state song!
B: That's pretty good. I didn't know that. I don't think my city really has a song.
Post: It's a nice song. Do you know who made it?

Knowledge Units:
(1) I love New York City and love its energy and diversity.
(2) The episode was written and directed by series creator Brad Falchuk, filmed in part on location in New York City, and first aired on May 24, 2011 on fox in the United States.
……

Target Response: No, I don't, but I searched the song. It was written and directed by Bard Falchuk.

Figure 1.2: *A three-turn dialogue example that consists of context, i.e., dialogue utterances, the corresponding knowledge sentences, and the target response. Note that the last context utterance is viewed as Post.*

This example illustrates several challenges. Firstly, how can we model contextual utterances, such as referring back to the 'song' in the context when generating responses? Secondly, how can we incorporate knowledge units into the generative model? The example presents multiple knowledge sentences that need to be included, posing a challenge in designing such a generative model. Thirdly, how can we effectively choose pertinent knowledge? It is apparent that the second knowledge sentence is more relevant than the first, so prioritizing sentences effectively is an important question. Furthermore, how can we filter out noise during knowledge infusion? When integrating knowledge sentences, most words other than the blue ones are irrelevant to the target response. Ignoring them is crucial to avoid affecting the response generation performance. Finally, how can we optimize the use of context for knowledge selection and response generation tasks? In the example, when performing knowledge selection, 'New York City' and 'song' should be emphasized as they help identify the correct knowledge sentence (the second one). However, during response generation, the context terms 'song' and 'good' should be prioritized. Thus, context needs to be used carefully when constructing a unified generative model that considers both context and knowledge. The detailed explanation of the five research questions will be introduced in the following section.

1.2 Research Questions

Automated generation of responses for open-domain dialogues presents challenges that have attracted significant interest from the research community. This has led to the research community's efforts to gather resources and formulate shared research tasks. For example, Dialog System Technology Challenge 7 (DSTC-7, Yoshino et al.) introduced a knowledge-grounded dataset derived from Reddit[5]. Furthermore, Dinan et al. (2019) developed a human-human dialogue dataset called Wizard of Wikipedia, annotated by Amazon Mechanical Turk (AMT) workers. Li et al. (2017b) published a dataset featuring topics from daily conversations, also labeled by employing AMT workers. Moghe et al. (2018) released the Holl-E dataset, where two speakers engage in a conversation about movies, with access to background knowledge about the films.

[5] https://www.reddit.com/

These shared resources have provided foundations for fruitful research and led to outstanding results. Nonetheless, there are still many challenges to tackle. In this section, I will discuss research questions (RQ) that are the primary focus of my work and illustrate the contributions for addressing these research questions. Table 1.1 is shown to illustrate the importance of these questions based on relevant literature and the discussion elaborates on the strategies that have been taken to approach these research questions in the related work.

RQ 1: Context Usage in Dialogue Generation

Ever since the sequence-to-sequence (Seq2Seq) model was proposed by Cho et al. (2014b), it rapidly became a popular architecture for solving natural language processing problems, such as machine translation tasks and dialogue generation tasks. Bahdanau et al. (2014) promoted basic RNN models with an attention mechanism whose decoder processes context using weighted scores derived from the dot-product of context and decoding token. The subsequent architecture, Transformer (Vaswani et al., 2017), directly used attention for both the encoder and decoder, boosting the performance of the generation task. As for the dialogue context, the researchers used the utterances from the conversational history.

As part of the context-aware dialogue generation, Serban et al. (2016) developed a hierarchical encoder-decoder framework (HRED) that utilizes two layers of RNNs to extract both utterance-level and dialogue-level vectors. The first layer of RNNs generates utterance-level vectors, while the second layer computes a dialogue-level vector using the last hidden state as the final vector. HRED establishes inter-relationships between utterances implicitly through the gates in the RNN. To improve upon HRED, Tian et al. (2017) proposed WSeq, which assigns weights to utterances based on their similarity with the post (i.e., the last utterance), thus explicitly weighting the importance of each utterance. Building on HRED, Xing et al. (2018) proposed a hierarchical

RQ 1: Context Usage in Dialogue Generation
How to affect response generation by incorporating intrinsic dialogue characteristics, such as speaker role, word prominence, and part-of-speech of content terms? Khandelwal et al. (2018) suggests that dialogue modeling should consider the dialogue's intrinsic characteristics, such as speaker-role and part-of-speech information. The initial dialogue generation research omitted these aspects (Serban et al., 2016; Xing et al., 2018; Zhang et al., 2019a; Serban et al., 2017) while the later incorporated single dialogue characteristics, e.g., speaker role embedding (Bao et al., 2020) or part-of-speech embedding (Sundararaman et al., 2021). The method proposed enables the use of multiple dialogue characteristics based auxiliary tasks (specific to an individual dialogue characteristic), used for supervised learning of embeddings and parameters of pre-trained models.

RQ 2: Knowledge Injection in Dialogue Generation
How to optimize injection of post-retrieved knowledge considering that such knowledge can be non-relevant to the response and introduce noise? In prior research, various backbone architectures, including RNN-based models (See et al., 2017), memory network-based models (Ghazvininejad et al., 2018), and Transformer-based models (Dinan et al., 2019), have included knowledge injection. I optimize the knowledge injection by differentiating knowledge sentences that are relevant to the post and expanding TED architecture to incorporate multiple knowledge sources.

RQ 3: Knowledge Selection in Dialogue Generation
How can knowledge be used considering its similarity to the post and best post-retrieved knowledge? Research has shown that incorporating all provided knowledge can lead to reduced performance due to the inclusion of noise (Zheng and Zhou, 2019). Some researchers have investigated the simultaneous training of knowledge selection and dialogue generation tasks (Lian et al., 2019; Kim et al., 2020; Dinan et al., 2019). Nevertheless, these knowledge selection modules are inherently tied to a particular generative model and cannot be used by other models. I designed a method which can decouple knowledge selection from response generation, allowing the knowledge to be ranked according to its similarity to the post and the best post-related knowledge.

RQ 4: Term-Level Knowledge De-noising in Dialogue Generation
How to eliminate term-level noise from knowledge injection considering that response is unknown and cannot be used to determine relevant knowledge during dialogue generation? Past research explored knowledge selection at the sentence level, whereby the selection of a knowledge unit (either a paragraph or a sentence) is used for knowledge injection (Zheng and Zhou, 2019; Ghazvininejad et al., 2018; Tam, 2020a). An important question is whether considering and de-noising knowledge units at a term level would be beneficial for dialogue generation. I introduced a method that prioritizes knowledge terms by reducing noise probability and emphasizing valuable knowledge terms. It uses post-retrieved knowledge to derive the Simulated Response Vector (SRV) via MLPs during training and use them as a ground-truth proxy during response generation.

RQ 5: Differentiating Context Use for Knowledge Selection and Response Generation
How to differentiate the role of context in knowledge selection and response generation using auxiliary tasks? I have reviewed context-aware architectures and knowledge injection models at both the sentence and term levels (Zheng and Zhou, 2019; Zheng et al., 2021). The context information is used for the knowledge selection and response generation sequentially, without the flexibility to be optimized for each of these two tasks. I introduce a CKL model, built upon the pre-trained BART model, which enables versatile context use based on context and contextualized knowledge vectors and derives three sets of weights for knowledge selection and response generation.

Table 1.1: Research questions (RQ) in context-aware and knowledge-grounded dialogue generation, formulated based on literature review and research explorations.

architecture that not only explicitly assigns weights to utterances but also to each word within the utterances. By performing a weighted-sum operation on the words, the utterance representations are obtained, and then another weighted-sum is performed on all utterances to produce the final dialogue representation.

The performance of context-aware generation models is further improved by Zhang et al. (2019a) through the use of Transformer architecture that employs self-attention to all utterances. At the same time, in order to further understand how context information is used by different architectures, Khandelwal et al. (2018) and Sankar et al. (2019a) conducted analyses on LSTM and Transformer models, respectively. Khandelwal et al. (2018) found that content words (i.e., nouns, verbs, and adjectives) are more important than function words, and frequent words require more context information during decoding. Sankar et al. (2019a) discovered that the Transformer-based model is not sensitive to word order and that the post, or the most recent utterance in the conversation, is the most important. Building on these findings and the idea that speakers tend to repeat themselves in a conversation (Clark and Wasow, 1998), the speaker role is considered an essential feature for improving a conversation model. Investigating effective ways to incorporate dialogue characteristics, e.g., speaker role and PoS, become challenging topics.

RQ 2: Knowledge Injection in Dialogue Generation

The origins of knowledge injection can be traced back to the DSTC-7 task, where a knowledge-grounded dataset was provided, derived from Reddit [6]. DSTC-7 researchers aimed to investigate how to incorporate knowledge and diversify generated responses. Ghazvininejad et al. (2018) employs a multi-task learning framework with three different encoders and a single decoder, to inject knowledge information based on the encoder and decoder parameter training. Tam (2020a) propose to adapt copy-mechanism to enrich responses

[6] https://www.reddit.com/

with copied knowledge and employ a cluster-based beam search to expand the output vocabulary space, thereby improving response diversity. Weston et al. (2018) suggests retrieving from the knowledge set by using the dialogue context as a query. The context and retrieved knowledge are concatenated and truncated to meet the input length requirement for constructing the generative model. Similarly, Yang et al. (2018) employs a retrieval mechanism to obtain candidate responses by computing the semantic similarity between context and given documents via a convolutional neural network, then expanding additional terms from the top-ranking documents of the retrieval model to the context. However, these generative models have limitations when incorporating knowledge: (1) they overlook the fact that different knowledge holds different significance, and (2) they do not control the amount of knowledge that is utilized while injected knowledge may include both useful information and noise. The answers to those questions are still unclear.

RQ 3: Knowledge Selection in Dialogue Generation

Research datasets used for dialogue generation typically include accompanying knowledge sets with potentially useful information. Research on knowledge injection leads to the conclusion that knowledge needs to be carefully selected (Zheng and Zhou, 2019). The problem of knowledge selection has been tackled in various ways. For example, Weston et al. (2018) retrieves knowledge using traditional information retrieval techniques and selects the top-ranked knowledge sentences for generating responses. Lian et al. (2019) proposes a method to approximate the parameters of a prior network and a posterior network to select knowledge related to posts and responses. Similarly, Kim et al. (2020) combines knowledge selection with generative models by treating it as a sequential decision problem. Zhao et al. (2020b) uses pre-trained models for response generation and uses an LSTM to assign scores to knowledge sentences dynamically and concatenate the top-scored knowledge with the context. Such knowledge selection modules are an integral part of the specific generative models and cannot be used by other models. Furthermore, the knowledge selection results are specific to the model and cannot be explained generally. The solution of this research question will be targeted in this thesis.

RQ 4: Term-Level Knowledge De-noising in Dialogue Generation

Knowledge selection can be done at different levels, e.g., paragraph, sentence, and term levels. As introduced, previous studies Zheng and Zhou (2019) treat knowledge terms equally, and Kim et al. (2020); Lian et al. (2019) preselect knowledge at the sentence level. Zheng et al. (2020) designed a retrieval paradigm to re-rank the knowledge sentences so that the generative models can obtain the relevant information. All of the selection-based methods have proven that correctly recognizing relevant knowledge information can help improve generative models' performance. Drawing ideas from this, I propose to view knowledge selection from a different angle, i.e., filtering noise from the term level. To my knowledge, it is still unexplored by the time when this work was done.

RQ 5: Differentiating Context Use for Knowledge Selection and Response Generation

A unified dialogue generation model considers both context and knowledge information. From the previous literature, we know that there are many effective unified generative models, such as Zhao et al. (2020b); Ghazvininejad et al. (2018); Zheng et al. (2021); Kim et al. (2020); Lian et al. (2019). However, those methods focus more on knowledge selection and knowledge injection and less on using the context information. The context has been primarily used for selecting knowledge to be combined with the same context to produce responses. Moreover, pre-trained models can also be seen as unified generative models. Lewis et al. (2020) proposed a new de-noising method (BART) to train the model rather than just using the masked language model (like the BERT Kenton and Toutanova (2019) does). Unlike the BERT and BART, Radford et al. directly pre-train the decoder as a large language model by receiving any length of input sequences. The pre-trained models focused on the effectiveness of leveraging large-scale datasets and did not care much about what the inputs were. As dialogue generation models, all of the previously mentioned models get both context and knowledge involved, however, they ignore the context's functionality should be different when conducting knowledge selection and response generation tasks. The way of context-differentiation will be investigated in this thesis.

1.3 Contributions

Regarding the research topics mentioned earlier, separate investigations were conducted, and approaches to address each of them were put forth. Within this section, the contributions outlined by the research questions will be presented.

1. RQ 1 was addressed by augmenting a pre-trained model with speaker role embeddings and part-of-speech embeddings to capture speaker role and part-of-speech information. Additionally, four auxiliary tasks were introduced to the generative model to provide explicit supervision for speaker role identification, post-word indication, part-of-speech recognition, and frequent-word classification. The proposed model GMATs, developed through this approach, outperformed seven strong baseline models.

2. To address the knowledge-injection issue (RQ 2), the Transformer with Expanded Decoder (TED) model was proposed. This model assigns different weights to knowledge on the decoder side, treating distinct and diverse knowledge units differently. Extensive experiments were conducted using varying numbers of knowledge sentences to determine the appropriate number of knowledge units for optimizing performance. The results indicate that a generative model should weigh knowledge units to attain optimal performance rather than injecting all the knowledge indiscriminately. For instance, the best number of knowledge units differs for different models (e.g., for the TED

model, the best number for the Wizard of Wikipedia dataset is 3, while for WSeq-Sum is 12).

3. In terms of RQ 3, a two-stage pipeline was proposed to separate knowledge selection from the response generation task. In the first stage, a knowledge selection model, TPPA, was constructed. This model utilizes two sets of knowledge distributions: response-retrieved knowledge and post-retrieved knowledge. These two distributions are fitted through a Transformer-based network to enable the post-related network to learn from the response-related network. Subsequently, the post-related network can assign scores to knowledge units, which can be used for retrieval during the test phase. In the second stage, the retrieved knowledge is applied to existing generative models, demonstrating the effectiveness of the proposed knowledge selection mechanism for specific generative models.

4. RQ 4 relates to term-level de-noising for the injected knowledge. The proposal prioritizes terms for selection and injection by decreasing the probability of noise and giving higher importance to terms that appear in responses. To that effect, the Knowledge Term Weighting Model (KTWM) was introduced to perform term-level de-noising of selected knowledge. The KTWM generates Simulated Response Vectors (SRVs) to mimic the response representation and uses SRVs along with the knowledge terms' embeddings to determine the weights of each knowledge term. This optimization has proven effective in learning the importance of each term and distinguishing relevant and non-relevant terms by weighing the useful terms higher for the purpose of response generation.

5. In relation to RQ 5, a novel Contextual Knowledge Learning (CKL) method was introduced to differentiate the context functionality and build a unified dialogue generation framework. The context is used in two ways: for the knowledge selection task and separately for the response generation task. To achieve this, Latent Weights with trainable latent vectors were introduced, which can be appropriately trained for each task. The model tuning is guided

through Contextual Knowledge Learning that begins with a Context Latent Weight Vector and a Contextualized Knowledge Latent Weight Vector. These vectors facilitate differential scoring of context utterances and knowledge sentences during the training process while explicitly capturing the relationship between context and knowledge. Experimental results show that CKL produces higher performance scores than 6 strong baselines, even when trained with only half of the dataset.

1.4 Thesis Outline

This dissertation focuses on dialogue generation methods that utilize both context and knowledge information. Chapter 1 outlines five key research questions related to dialogue generation and presents experiments conducted to address these questions. Chapter 2 introduces prevalent practices in the field, including backbone architectures such as RNNs, Transformers, and pre-trained models (e.g., BERT, BART). It also covers evaluation metrics like BLEU, Meteor, Rouge, Embedding Average, Greedy, Extrema, BERT-score, and distinct scores, and discusses datasets such as Wizard of Wikipedia, Reddit, Holl-E, CMU-DoG, and DailyDialog. Chapter 3 offers a comprehensive literature review for each research question, providing context for my research in relation to traditional dialogue generation models and the latest state-of-the-art methods.

Chapters 4 to 8 present the proposed approaches for addressing the five research questions. In Chapter 4, the generative model GMATs is introduced, leveraging intrinsic dialogue characteristics like speaker roles and part-of-speech for context projection. Chapter 5 introduces the TED model for knowledge injection, assigning importance scores to individual pieces of knowledge. Chapter 6 focuses on knowledge selection, presenting the TPPA model, which approximates post-related knowledge to response-related knowledge distributions, prioritizing relevant knowledge during testing. In Chapter 7, a term-level knowledge de-noising mechanism (KTWM) is devised to address non-relevant knowledge term issues. Finally, Chapter 8 proposes the unified generative model CKL, incorporating both context and knowledge information, with distinct usage of context for knowledge selection and response generation. Chapter 9 provides a summary, concluding remarks, and discusses potential future directions for dialogue generation research.

Chapter 2

Background

To be self-contained, in this Chapter, I will introduce the key parts of the (1) backbone of the generative models, (2) metrics for dialogue generation, and (3) common benchmarks which are used in the experiments.

2.1 Backbones of the Generative Models

Dialogue generation techniques have evolved from recurrent neural networks to Transformers and, more recently, pre-trained models. The advancement of these techniques has propelled the development of dialogue generation models. In this section, three commonly used backbones for dialogue generation will be discussed: Recurrent Neural Networks, Transformers, and pre-trained models.

2.1.1 Recurrent Neural Networks

The recurrent neural network (RNN, Rumelhart et al. (1986)) is a type of neural network that is designed with a cycle of connections between nodes, enabling the output from some nodes to affect subsequent input to the same nodes. This characteristic of RNN allows it to receive variable-length sequences of inputs and capture all the information of the inputs in the RNN parameters. To make the RNN more effective, two widely used variants, namely Long short-term memory (LSTM) and Gated recurrent units (GRUs) have been proposed. LSTM, as proposed by Gers et al. (2000), is composed of a cell, a forget gate, an input gate, and an output gate. The cell remembers values over arbitrary time intervals, while the three gates regulate the flow of information into and out of the cell. The forget gates determine what information to discard from a previous state by assigning a weight, input gates decide which pieces of new information to store in the current state, and output gates control which pieces of information in the current state to output. This way, an RNN can store sequence information for any length, hence the name "Long short-term memory." GRUs, proposed by Cho et al. (2014a), are similar to LSTM but have fewer parameters as they lack an output gate. Due to their recurrent characteristic, LSTM and GRU are widely used in processing text and performing natural language processing tasks such as machine translation and dialogue generation.

2.1.2 Transformer

The Transformer, introduced by Vaswani et al. (2017), is a deep learning model comprising a self-attention layer, a cross-attention layer, and a feed-forward layer, each enclosed in a residual network. The self-attention layer is the main component that assigns varying levels of importance to different segments of the input data. [1]. In the Transformer, the self-attention mechanism computes semantic similarities between the current word and all the other words in the input sequence. Unlike traditional methods that use a dot product to calculate attention, the Transformer uses a scaled dot product, where the dot product is divided by $\sqrt{d}$, representing the dimension. The attention equation is presented below.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{2.1}$$

[1] https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)

where, Q, K, and V are matrices which are a set of queries, keys, and values respectively. Unique to the transformer, multi-head attention is employed. Instead of using the d dimension to compute attention, multi-head attention divides d into several sub-dimensions. This allows each word to access multiple sub-spaces of the other words when generating a word representation. After calculating attention at different heads, the Transformer combines all sub-space attentions again for the next step.

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O \tag{2.2}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{2.3}$$

where W^O, W_i^Q, W_i^K, W_i^V are trainable parameters. Recently, the Transformer architecture has become a crucial building block for state-of-the-art pre-trained models, including BERT(Kenton and Toutanova, 2019), BART(Lewis et al., 2020), GPTs(Brown et al., 2020), and T5(Raffel et al., 2020).

2.1.3 Pre-trained Models

The Transformer-based pre-trained models that are popular today can be traced back to the Bidirectional Encoder Representations from Transformers (BERT) proposed by Kenton and Toutanova (2019). [2].

BERT is based on the Transformer architecture and is composed of several Transformer encoder layers. It is pre-trained on two tasks: language modeling and next-sentence prediction. To accomplish this, 15% of the input tokens are randomly masked, and the objective is to recover the original token given the masked input sequences. Additionally, the model is trained to classify whether two given sequences appear sequentially in the training corpus. Through this training process, BERT learns latent representations of words and sentences in context. Once pre-trained, BERT can be fine-tuned with fewer resources on

[2] https://en.wikipedia.org/wiki/BERT_(language_model)

smaller datasets to improve its performance on specific tasks like text classification and dialogue generation.

In this work, I adopt the pre-trained model BART (Lewis et al., 2020) which is elaborately designed for text generation. BART is a de-noising autoencoder for pre-training sequence-to-sequence models. It is trained by (1) corrupting text with an arbitrary noising function, and (2) learning a model to reconstruct the original text. The author experiments with several previously proposed and novel transformations for the BART model, including (a) Token Masking (like the BERT, masking out tokens from the input sequence), (b) Token Deletion (random tokens are deleted from the input), (c) Text Infilling (a number of text spans are sampled, with span lengths drawn from a Poisson distribution with $\lambda = 3$), (d) Sentence Permutation (a document is divided into sentences based on full stops, and these sentences are shuffled in random order), (e) Document Rotation (a token is chosen uniformly at random, and the document is rotated so that it begins with that token). Through this setting, BART generalizes the original word masking and next sentence prediction objectives in BERT by forcing the model to reason more about overall sentence length and make longer-range transformations to the input. A set of experimental evidence is also provided in the original paper to illustrate the advantage of BART, especially on text generation tasks. There are also other excellent pre-trained models, such as RoBERTa(Liu et al., 2019), XLNet(Yang et al., 2019), T5(Raffel et al., 2020), GPT-3(Brown et al., 2020). Please refer to the corresponding papers if needed.

2.2 Evaluation Metrics

A set of evaluation metrics used to assess the performance of dialogue generation models will be described in this section. Previous studies have explored different measurement techniques, which fall into two main categories: lexical metrics and embedding-based metrics. Lexical measurements focus on the similarity between two sequences in terms of n-grams, which include BLEU, Meteor, and Rouge. On the other hand, embedding-based metrics leverage pre-trained embedding models like Glove (Pennington et al., 2014) and BERT to measure the semantic similarity between two sequences. To measure the diversity of generated responses, a diversity score is introduced, which will be discussed in the following sub-sections.

2.2.1 Lexical Metrics

1. BLEU

The origin of BLEU can be traced back to the machine translation task, where the quality of a translated sequence needs to be compared to the reference language sequence. Human evaluation, although considered the gold standard, is slow, costly, and often subject to inconsistencies due to individual biases. To overcome these challenges and accelerate research, Papineni et al. (2002) proposed BLEU (Bilingual Evaluation Understudy) as an automated metric for evaluating machine translation quality. BLEU calculates the modified n-gram precision by comparing the n-grams in the candidate translation with those in the reference translations, with higher n-gram values focusing on longer and more complex matches. The precision scores for different n-gram orders are then combined using a weighted geometric mean. To penalize overly short translations, BLEU also incorporates a brevity penalty factor, which adjusts the final score based on the length ratio between the candidate translation and the closest reference translation length. In other words, higher BLEU scores indicate greater similarity between two sequences, with more common words and phrases contributing to a higher score. The metric has been widely adopted in the field of machine translation due to its simplicity, ease of use, and correlation with human judgment. Recently, it also become widespread in the dialogue generation domain.

2. Meteor

BLEU primarily focuses on word-level precision, which can be limiting when the generated sequence is accurate but not exactly the same as the reference. To address this issue, Banerjee and Lavie (2005) introduced Meteor (Metric for Evaluation of Translation with Explicit ORdering), which incorporates not only lexical words but also synonymy and word stems through the use of WordNet. By considering both lexical and semantic information, Meteor provides a more comprehensive evaluation of the generated sequence against the reference. According to studies by Liu et al. (2016a) and Liu et al. (2021e), BLEU-2 and Meteor are more correlated with human evaluation, with Meteor generally outperforming BLEU-2. These findings indicate that considering semantic information in addition to lexical information can lead to better alignment with human judgment. Additionally, Liu et al. (2021e) suggest that embedding-based metrics, which leverage pre-trained word embeddings to measure semantic similarity, perform better than most lexical-based metrics. The development of alternative evaluation metrics, such as Meteor, highlights the importance of incorporating semantic information when assessing the quality of machine-generated text. By using a combination of lexical and semantic features, researchers can obtain a more accurate understanding of the performance of their models, ultimately leading to improvements in natural language processing technologies.

3. Rouge

Following the emergence of neural machine translation models, generative models have the ability to produce fluent responses. However, there are cases where the translation models may perform partial translation by omitting some content of the input. As a solution, the Rouge metric was introduced by Lin (2004) with the aim of evaluating the quality of a sequence automatically by comparing it to a human-created reference sequence. The Rouge metric consists of several variants, such as Rouge-N, Rouge-L, and Rouge-S. Rouge-N evaluates the n-gram overlap between the generated and reference sequences, with higher n-gram values focusing on longer, more complex matches. Rouge-L, on the other hand, measures the longest common sub-sequence (LCS) between the two sequences, emphasizing the importance of continuous matches. Rouge-S evaluates the skip-bigram co-occurrence, allowing for flexibility in word order while still considering semantic similarity.

2.2.2 Embedding-based Metrics

1. Embedding-based Measurement

Instead of relying on lexical information, an alternative approach for dialogue evaluation is to use embedding-based metrics that consider the semantic meaning of each word. These metrics are based on word embeddings, which assign a vector to each word. Three main methods of embedding-based metrics are described in Liu et al. (2016a): Embedding Average (Mitchell and Lapata, 2008), Greedy Matching (Rus and Lintean, 2012), and Vector Extrema (Forgues et al.).

Embedding Average calculates the cosine distance between sentence-level representations (such as the predicted and ground truth responses) by averaging the constituent word embeddings and then taking the average across all pairs. The Greedy Matching metric, on the other hand, finds the maximum cosine scores in the similarity matrix by considering both rows and columns. The Vector Extrema metric creates a sentence vector for each of the two sentences by selecting the highest word-embedding values along each dimension and then calculates the similarity score between the two vectors.

2. BERT-Score

Since the BERT (Kenton and Toutanova, 2019) was proposed, the word embeddings can be obtained by the pre-trained models. Zhang et al. (2019b) introduce the BERT-Score metric as a means of evaluating text generation based on word representations, utilizing the pre-trained BERT model. They highlight two common shortcomings of n-gram-based metrics such as BLEU and Meteor. Firstly, these metrics often struggle to accurately match paraphrases as they only count overlapped words rather than considering semantic information. This can result in a reduction in performance when semantically correct phrases are penalized due to differences from the surface form of references. In contrast, BERT-Score calculates similarity using contextualized token embeddings, which can capture word information in context. Secondly, n-gram models may fail to account for distant dependencies and can penalize important changes in word order. In contrast, contextualized embeddings are designed to capture such dependencies and effectively represent word order.

2.2.3 Diversity Metric - Distinct Scores

While the aforementioned metrics are useful for evaluating the similarity between two sequences (one being the generated sequence and the other being the ground truth reference), they do not provide insight into the informativeness of the generated sequences, specifically the diversity of the produced text. To address this issue, Li et al. (2016a) propose using n-gram distinct scores to evaluate whether the generated sequences utilize a variety of vocabularies and avoid repetitive language. The commonly used practices involve using uni-gram and bi-gram for evaluation, which are denoted as ‘Div-1’ and ‘Div-2’, respectively. For instance, the Div-1 score is computed by dividing the number of distinct uni-gram words by the total number of generated words. Similarly, the Div-2 score is calculated by dividing the number of distinct bi-grams by the total number of generated bi-grams. By assessing the diversity of the generated text, these scores provide a measure of the model’s ability to generate informative and varied content.

2.3 Datasets

In this section, all datasets that are used in the thesis will be illustrated, including Wizard of Wikipedia, DailyDialog, Reddit, Holl-E, and CMU-DoG datasets. Four of them contain human annotation except for the Reddit which is extracted from the Reddit forum. These datasets are selected because are widely adopted in the prior works and are publicly available. The detailed statistics are demonstrated in Table 2.1.

Table 2.1: Statistics of each dataset used in this thesis. 'k' means thousand, and 'M' denotes million.

Data sets	Train/Validation/Test Size	Used in Chapters	Human Annotation
Wizard of Wikipedia	74k/3.9k/3.8k	Chapters 4, 5, 6, 7, and 8	Yes
DailyDialog	12.8k/1k/0.9k	Chapters 4	Yes
Reddit	3M/not provided/13k	Chapter 5	No
Holl-E	34k/4.3k/4.3k	Chapters 6 and 7	Yes
CMU-DoG	6.6k/3k/10.5k	Chapter 8	Yes

2.3.1 Wizard of Wikipedia

The Wizard of Wikipedia dataset (also termed WoW in this thesis), is a dataset developed by Facebook AI Research, as described in Dinan et al. (2019). It comprises a collection of context, responses, and retrieved background knowledge. The authors used the Amazon Mechanical Turk (AMT) platform[3] to crowdsource a diverse range of 1365 natural, open-domain conversational topics, including but not limited to commuting, Gouda cheese, music festivals, podcasts, bowling, and Arnold Schwarzenegger. In this dataset, there are two human roles: the Apprentice and the Wizard. The Wizard answers the Apprentice's questions based on either the retrieved knowledge of the last utterance or their own knowledge, meaning that all answers are generated by humans.

The dataset is divided into train, validation, and test sets. For the validation and test sets, two versions exist based on the topic's presence in the train set: a seen set (topics present in the train set) and an unseen set (new topics

[3] https://www.mturk.com/

absent in the train set). The original dataset can be accessed through ParlAI[4]. The sizes of the train/seen validation/seen test sets are 74,092/3,939/3,865. For the unseen validation/unseen test sets, the sizes are 3,927/3,924.

2.3.2 DailyDialog

DailyDialog is a multi-turn dialogue data set developed by Li et al. (2017b). The topics in this dataset come from daily communication and because of this, the author claims four special characteristics of this dataset. (1) Daily Topics: It covers ten categories ranging from ordinary life to financial topics, which is different from domain-specific datasets. (2) Bi-turn Dialog Flow: It conforms to basic dialogue act flows, such as Questions-Inform and Directives-Commissives bi-turn flows, making it different from question-answering datasets and post-reply datasets. (3) Certain Communication Patterns: It follows unique multi-turn dialogue flow patterns reflecting human communication style, which are rarely seen in task-oriented datasets. (4) Rich Emotion: It contains rich emotions and is labeled manually to keep high quality, which is distinguished from most existing dialogue datasets. The DailyDialog dataset [5] is split into train/validation/test sets. I found there are some overlapping samples between the train set and the test set, so the samples that appeared in both the train and test set are removed. After pre-precessing the data, the sizes of the train/validation/test sets are 12,844/1,081/974.

2.3.3 Reddit

This conversational data set is released by Dialog System Technology Challenges 7 (DSTC-7) (Yoshino et al.) and is extracted from Reddit. Each conversation is typically initiated with a URL to a web page (grounding) that defines the subject of the conversation. To reduce spamming and offensive language and improve the overall quality of the data, the author manually whitelisted

[4] http://parl.ai

[5] http://yanran.li/dailydialog

the domains of these URLs and the Reddit topics, i.e., “subreddits”, in which they appear. This filtering yielded about 3 million conversational responses and 20 million facts divided into train, validation, and tests. For the test set, in order to provide multi-reference for evaluation, the author filtered the dataset by limiting the conversational turns to 6 or more responses. There are 13,440 samples in the test set in the end. I download the data from Reddit dump and Common Crawl [6] as the experiment data, following DSTC-7[7].

2.3.4 Holl-E

Moghe et al. (2018) build a dataset, Holl-E, based on a hypothesis that it is common for humans to produce responses by copying or suitably modifying from the background knowledge. Similar to the WoW dataset, the authors ask the AMT workers to chat about a movie using various sources as background knowledge, e.g. the plots, reviews, comments, and fact tables. The response to a post is either copied or suitably modified from the grounded knowledge, which means there are a number of responses that exactly exist in the background knowledge. In total, it contains around 9,000 conversations with 90,000 utterances, covering about 921 movies. There are three settings of the original dataset and in this thesis, the mixed-long setting is used because it is the most complicated one and the others are sub-sets of it.

2.3.5 CMU-DoG

The CMU-DoG dataset, proposed by Zhou et al. (2018), involves workers from AMT and focuses on conversations about movies using Wikipedia articles as background documents. The dataset creation involves two workers exchanging ideas, with one worker having access to the document and the other without access in one scenario, and both workers having access in another scenario.

[6] http://files.pushshift.io/reddit/comments/, http://commoncrawl.org/

[7] https://github.com/mgalley/DSTC7-End-to-End-Conversation-Modeling/tree/master/data_extraction

The workers are given instructions to chat for at least 12 turns. The original dataset is available from the published paper [8] and contains 4,112 conversations with an average of 21.43 turns per conversation. Additionally, Li et al. (2019b) released a tokenized version of the CMU-DoG dataset, which will be utilized in this thesis. The train/validation/test sets consist of 66,332/3,269/10,502 samples.

2.4 Key Notations

In this thesis, the symbol D is used to refer to a dialogue that comprises two speakers, a "poster" and a "responder", exchanging utterances u_i as part of a context (C) and a response (R). The context consists of n utterances, $C = u_1, u_2, \ldots, u_i, \ldots, u_n$, where u_i denotes the i-th utterance, and the last utterance (u_n) represents the post. For each context-response pair, there is a set of knowledge K, which contains several sentences, denoted as $K = k_1, k_2, \ldots, k_j, \ldots$, where k_j represents the j-th knowledge sentence. These notations are summarized in Table 2.2. Moreover, I define the term "useful word" to be the word that appears in both the response and the corresponding knowledge (external knowledge article) but not in the context. This definition will be employed for analyzing the experimental results. The other specific notations will be introduced as required in each section.

Table 2.2: Summary of the key notations used in the thesis.

Notations	Description
D	Dialogue Conversation
C	Input Context
K	Input Knowledge
R	Response
u_i	i-th utterance of the context
k_j	j-th sentence of the knowledge

[8] https://github.com/festvox/datasets-CMU_DoG

Chapter 3

Related Work

In this Chapter, I conduct a literature review of various topics related to generative models, which include sequence-to-sequence models, dialogue characteristics, context-aware dialogue generation, knowledge-grounded dialogue generation, knowledge selection, and novel explorations for dialogue generation models. My focus is on how these topics intersect with the task of generating dialogue given a contextual history and prior knowledge.

3.1 Sequence-to-Sequence Models

The Seq2Seq model is responsible for transforming an input sequence into an output sequence, and numerous architectures have been proposed to generate conversational responses. In the past, RNNs have been the most popular generative model backbone, as described in Chapter 2. With the introduction of the attention mechanism (Bahdanau et al., 2014), RNN-based Seq2Seq models have become more prevalent. This mechanism allows the decoder to look back at all the words in the encoder, taking their similarities as weights to incorporate the input sequences. For instance, Luong et al. (2015); Wu et al. (2016) uses an RNN-based Seq2Seq model with the attention mechanism to complete the machine translation task. To further enhance the Seq2Seq model's performance, See et al. (2017) introduces a pointer network that leverages the copy mechanism (Gu et al., 2016) to decide whether to generate the current word or copy it from the context. The author notes that the generative mechanism is more critical than the copy mechanism, particularly at the beginning of the generative process when faced with uncertainty. Building on See et al. (2017), Tam (2020b) incorporates external knowledge by creating a dual copy mechanism. When the decoder generates a new word, it checks whether the word comes from the generative model, the context, or the external knowledge.

Unlike the RNN-based Seq2Seq model, the Memory Neural Networks (MemNNs, Weston et al. (2015)) can reason through the inference component by accessing long-term memory. Sukhbaatar et al. (2015) proposes an end-to-end MemNNs model, which injects memories into a query vector from the bag-of-words representation of both the query and memories. Li et al. (2017a) adopts an offline supervised method and an online reinforcement learning-based method to use the MemNN as a generative model for response prediction. They also create a MemNN binary classifier to determine which sequence to respond to. Many other previous studies, including (Bordes et al., 2017; Wang et al., 2018; Ma et al., 2017), have used MemNNs as the backbone of dialogue models.

The Transformer (Vaswani et al., 2017) includes both self-attention and cross-attention for its encoder and decoder to fully utilize the attention mechanism. Due to its exceptional performance, numerous variants have been proposed on top of the original Transformer. Dehghani et al. (2019c) introduces an improvement by incorporating the Adaptive Computation Time (ACT, Graves (2016)), which automatically determines the number of steps to run for a word representation instead of using a fixed number of steps. Transformer-XL (Dai et al., 2019) is capable of capturing longer-term dependencies that the original Transformer cannot and is over 1800 times faster than the vanilla version.

3.2 Dialogue Characteristics

Refining response generation techniques can benefit from an understanding of human-human dialogue characteristics. According to Clark and Wasow (1998), individuals have a tendency to repeat words used during a conversation. This finding suggests that identifying the *speaker roles* could aid in identifying relevant vocabulary for response generation. The use of speaker roles has proven effective in response-retrieval tasks (Liu et al., 2021a; Zhang et al., 2018a). For generative methods, Rashkin et al. (2021) incorporates speaker role prompts by manually adding controllable tags to a pre-trained language model. In contrast, Bao et al. (2020) proposes embedding the speaker role and merging it into the word representations.

Experiments conducted by Khandelwal et al. (2018) aim to investigate how generative models such as LSTM utilize context information. The results reveal that content words, which include nouns, verbs, and adjectives, are more important than function words, while frequent words require more context information than infrequent ones when making predictions. In a separate study, Sankar et al. (2019a) explores how the Transformer employs context information and finds that word order does not significantly impact its performance, except for the last utterance in the dialogue context, which has a considerable influence on response prediction. Given the demonstrated usefulness of these dialogue characteristics in previous literature, it is worthwhile to investigate incorporating them into the generative models.

3.3 Context-Aware Dialogue Generation

As discussed in Sec. 3.2, the speaker's role, content words, and frequent words are significant factors in enhancing dialogue generation. Numerous earlier works have examined these characteristics for context-aware dialogue generation tasks, where only context information is accessible to the generative model, independent of knowledge. I examine approaches to dialogue generation models that are solely related to context, utilizing utterance-level modeling, word-level modeling, and speaker-role considerations.

Utterance level modeling. The RNN-based Seq2Seq models consider the hidden state of the last step as a sentence-level representation. To improve upon this, the Hierarchical Recurrent Encoder-Decoder (HRED, Serban et al. (2016)) method creates a hierarchical system where the RNN generates a representation of each context utterance at the first level, and at the second level, the RNN takes utterance representations as input to create the entire context-level vector. However, due to the complex dependencies among sub-sequences of utterances that HRED cannot capture well, Serban et al. (2017) introduces the Latent Variable Hierarchical Recurrent Encoder-Decoder (VHRED) model, which enhances HRED with a stochastic latent variable at the utterance level to model the inter-relations among utterances. The Weighted Sequential (WSeq, Tian et al. (2017)) integration further improves upon this by giving weights to the utterances based on their similarity with the post and creating a context-level representation as a weighted sum of all utterances. The Hierarchical Recurrent Attention Network (HRAN, Xing et al. (2018)) assigns weights to both the utterance vectors and to the words within utterances. Relevant Contexts with Self-Attention (ReCoSa, Zhang et al. (2019a)) update utterance representations using the self-attention mechanism, where the utterance weights are calculated implicitly.

Word-level modeling. Zhao et al. (2020c) is influenced by the findings of Khandelwal et al. (2018) and considers several sub-tasks related to response generation, such as predicting word and utterance order simultaneously. Similar efforts are also motivated by Sankar et al. (2019a). Recently, topic models have been incorporated into response generation methods, such as the STAR-BTM Model (Zhang et al., 2020) that generates topic information for each utterance using the Biterm Topic Model (BTM, Yan et al. (2013)). However, because the topics in STAR-BTM are static, Ling et al. (2021) explores a dynamic context-controlled topic transition strategy that leverages contextual topics to obtain relevant transition words. This approach captures the relationship between context history and the next topic representation. As for incorporating part-of-speech (PoS) information, Niehues and Cho (2017) uses a multi-task learning scheme to predict PoS tags and target response in the meantime, and this work did not consider PoS embeddings. In contrast, Sennrich and Haddow (2016) creates a PoS embedding and concatenates it with the word embedding to form a new word representation. For dialogue generation, Li et al. (2019a) designs a PoS embedding matrix, added up with the normal word embedding to generate sentences.

Speaker-role considerations. Previous studies have shown the effectiveness of utilizing speaker-role information in response retrieval tasks Liu et al. (2021a); Ouchi and Tsuboi (2016); Zhang et al. (2018a), so many researchers have considered utilizing it in response generation tasks.

Ouchi and Tsuboi (2016) formalizes the task of addressee and response selection for multi-party conversations, where the speaker-role and response selection are modeled in the same framework. To address this task, Zhang et al. (2018a) proposes the Speaker Interaction Recurrent Neural Network (SI-RNN), which updates a sender embedding and an addressee embedding during the training process to distinguish between different speakers. The SI-RNN is then used to jointly perform addressee and response selection. Meanwhile, Liu et al. (2021a) creates an utterance-aware channel and a speaker-aware channel to update word and utterance representations using the mask trick. This approach decouples the speaker-role information from the dialogue history rather than treating the context as a whole.

When it comes to generative dialogue, several approaches have integrated speaker-role information in different ways. For instance, Bao et al. (2020) introduces a discrete latent variable to PLATO, a pre-trained model that addresses the diverse response problem. They embed the speaker-role information and add it to the word representation. Similarly, Wu et al. (2021a) utilizes a pre-trained model (GPT2, Radford et al.) to create a memory recurrence mechanism for storing previous dialogue utterances. In another work, Rashkin et al. (2021) adds manually controllable tags, including speaker role tags, to a pre-trained language model.

However, previous studies have incorporated speaker-role information without well-supervision, either for response selection or response generation. In this thesis, how to embed speaker-role information and supervise its training process will be operated.

3.4 Knowledge-Grounded Dialogue Generation

Since the release of the DSTC-7 knowledge-based dialogue generation task, interest in the topic has been increasing steadily. Ghazvininejad et al. (2018) proposes a multi-task learning approach that uses posts and knowledge in the encoders and shares the same decoder parameters. Luan et al. (2017) expands the scope by incorporating personality information into the model. They assumed that trainable parameters could potentially capture persona from non-conversational data such as Tweets. To include external knowledge in addition to the context, Yavuz et al. (2019) adopts pointer-generator networks within a hierarchical framework. Ye et al. (2020) proposes a latent variable-based generative model that contains a joint attention mechanism conditioned on both context and external knowledge. Li et al. (2019b) applies a deliberation network to create a two-stage generative model that combines both context and knowledge and, in the second generation stage, makes use of the outputs from the first stage. Zheng and Zhou (2019) proposes the Transformer with Expanded Decoder (TED) architecture that assigns different weights to different knowledge sources and incorporates them into the generation process. Recently, many generative models based on pre-trained models have achieved high performance. For example, Zhao et al. (2020b) uses BERT encoders and a GPT-2 decoder for knowledge projection and prediction. Liu et al. (2021c) leverages BART as the backbone to fine-tune the model with few-resource datasets. Prabhumoye et al. (2021) also chooses BART as the basic pre-trained framework to project context and knowledge in a unified model.

Research has shown that pre-trained models can achieve a better understanding of inputs when provided with appropriate prompts (Liu et al., 2021b). For example, prompts indicating the locations of context and knowledge in the input sequence can enhance generation performance. Zheng and Huang (2021) manually designs prompts for context, knowledge, and response to enable the GPT-2 language model to capture key information from each. However, this approach requires significant human effort for prompt design, motivating the use of continuous prompt representations. Li and Liang (2021) trains prompt embeddings in the latent space and concatenates them with the input sequence, eliminating the need for manual prompt design. Building on this, Liu et al. (2021d) incorporates a trainable prompt into the GPT model to improve understanding ability. Gu et al. (2021) generates prompt vectors by conditioning on the context and concatenating them with the context to generate responses. Nonetheless, despite the benefits of pre-trained models and prompt learning, knowledge selection remains useful for dialogue generation.

3.5 Knowledge Selection for Dialogue Generation

Although generative models can effectively incorporate knowledge, accurately identifying relevant knowledge is still beneficial. Weston et al. (2018) proposes a method to retrieve candidate content from a knowledge set and use it to refine the post. Lian et al. (2019) selects knowledge by approximating the posterior distribution using the prior distribution and then injecting it into the decoder. Kim et al. (2020) trains a knowledge selection module and a response generation module jointly, treating the knowledge selection as a sequential decision problem and utilizing input and knowledge from previous turns to select knowledge in subsequent turns. In this work, I propose a knowledge selection method and a noise-avoiding mechanism and integrate the knowledge de-noising method into a unified generative model that considers both context and knowledge.

3.6 Large Language Models (LLMs)

In Section 2.1.3, pre-trained models such as BERT, BART, and GPT are outlined, all of which are constructed using the Transformer architecture. These models incorporate multiple layers and employ multi-head attention mechanisms for calculation. Initially, they undergo pre-training on extensive datasets gathered from the Internet, demonstrating their superiority when compared to models trained from scratch. Recently, there has been a surge in the development of large language models, resulting in significant advancements in various natural language tasks, including text generation, comprehension, and translation. Essentially, these large language models are still pre-trained, but they feature more Transformer layers and a greater number of parameters. This trend began with the emergence of GPT-3 (Brown et al., 2020), and subsequent applications like ChatGPT (Ouyang et al., 2022) is built upon it, introducing a novel mechanism known as reinforcement learning from human feedback (RLHF). The largest GPT-3 model comprises a staggering 175 billion parameters, making it 116 times larger than its predecessor, GPT-2, which had 1.5 billion parameters.

The remarkable success of ChatGPT has led to the development of numerous Large Language Models (LLMs), as outlined in the survey by Zhao et al. (2023). Among these models, some well-known models include LaMDA, which contains 137 billion parameters, and PaLM (Chowdhery et al., 2022), which has even 540 billion parameters. Most recently, OpenAI has announced the release of GPT-4 [1]. However, specific details about the model, including its parameter size, have not been disclosed.

These large language models result in applications in diverse domains. In natural language understanding, they have been used for sentiment analysis, named entity recognition, and question answering. In machine translation, LLMs have improved translation accuracy. Furthermore, these models have demonstrated exceptional performance in text generation tasks, including chatbots, content creation, and code generation. Despite their remarkable achievements, large language models are not without challenges. Ethical concerns regarding bias in model outputs and environmental concerns due to their computational demands have raised important questions (Yan et al.). Additionally, fine-tuning large models for specific tasks often requires substantial data and computational resources.

3.7 Summary

In alignment with the research questions outlined in Chapter 1, a comprehensive review of the literature is presented to establish the basis for the research undertaken in this thesis. Here are the key findings and outcomes derived from the literature review:

Firstly, the fundamental sequence-to-sequence models (Section 3.1), such as the Transformer, serve as the building blocks for all subsequent research. While exploring the research on dialogue characteristics (Section 3.2) and context-aware dialogue generation (Section 3.3), a gap was identified in effectively linking dialogue characteristics with dialogue generation. As a result, the introduction of the GMATs model is discussed in Chapter 4.

The investigations in Section 3.4 emphasized the critical importance of incorporating knowledge. In Chapter 5, the TED model is introduced, infusing knowledge by assigning weights to each integrated knowledge sentence.

[1] https://openai.com/research/gpt-4

Recognizing that not all knowledge should be injected (in Section 3.5), as supported by findings in the literature, two aspects were examined: performing knowledge selection at the sentence level and refining injected knowledge at the term level. This led to the development of the TPPA model (Chapter 6) and the KTWM model (Chapter 7).

Lastly, the review of the aforementioned research revealed that they treat context as static information for simultaneously handling knowledge selection and response generation tasks. However, this approach may not align with real-world scenarios. To address this and differentiate the role of context, the CKL model was introduced in Chapter 8.

These notable studies lay the foundation for the advancement of more sophisticated dialogue-generation solutions, particularly in Section 3.6, where large language models generate remarkable results, opening up possibilities for the practical application of automated models. In the following chapters, individual methods to address each of the five research questions will be presented, respectively.

Chapter 4

Response Generation Method Given Context Information

In Section 3.2, exploration was done regarding the limited information available in the context for generating dialogues, emphasizing the necessity to investigate dialogue characteristics. Previous research has demonstrated that generating dialogues based solely on context is a challenging task, but the research community has been gradually making progress, particularly by focusing on two-speaker dialogues. These speakers are typically referred to as the ‘poster’ and the ‘responder,’ with the system generating the dialogue turn of the responder. Although the roles of the poster and responder may be defined differently in some cases, for this work, the last utterance in the dialogue sequence is considered the post corresponding to the ‘poster speaker,’ and the generated response is attributed to the ‘responder speaker,’ regardless of who initiates the dialogue.

A fundamental challenge in the dialogue generation is sourcing vocabulary relevant to the dialogue context that the system can utilize. In the context, all prior utterances exchanged by the speakers during their dialogue turns are considered. In a related area of research, context is combined with external knowledge (Ghazvininejad et al., 2018; Kim et al., 2020; Zheng et al., 2021) , where the primary challenge lies in selecting relevant knowledge and injecting it into the dialogue generation process.

This work primarily focuses on the context-only approach and presents fresh insights into using pre-trained models with auxiliary tasks to enhance response generation. Although BART has demonstrated its effectiveness as an architectural framework, determining a systematic approach for selecting relevant dialogue aspects to incorporate into the generative model remains a challenge. This research provides new perspectives on using pre-trained models in conjunction with auxiliary tasks to guide the learning of general and specific characteristics for particular dialogue aspects. The primary focus is on creating auxiliary tasks that align with the generative model and are effective in enhancing the learning of dialogue characteristics that contribute to response generation.

Building upon previous work by Khandelwal et al. (2018), which highlights the benefit of generative models focusing on content words characterized by specific Part-of-Speech (PoS) types (nouns, verbs, and adjectives), auxiliary tasks are designed to distinguish between general and information-rich content. Specifically, complementary auxiliary tasks are devised, one for frequent words and the other for content words. These tasks, with distinct loss functions during training, address both general and specific content aspects that the model may need to balance.

Similarly, Sankar et al. (2019a) revealed that the post, i.e., the last utterance of the poster speaker, significantly influences response prediction. As a result, auxiliary tasks are defined to facilitate the learning of speaker role indicators in general and post words in particular.

Two key research questions are considered:

- Whether pairing auxiliary tasks to cover *general* and *specific* characteristics can be beneficial?

- Which aspects, *general* or *specific* characteristics, would be more beneficial, at least for the dialogue content representation and the dialogue

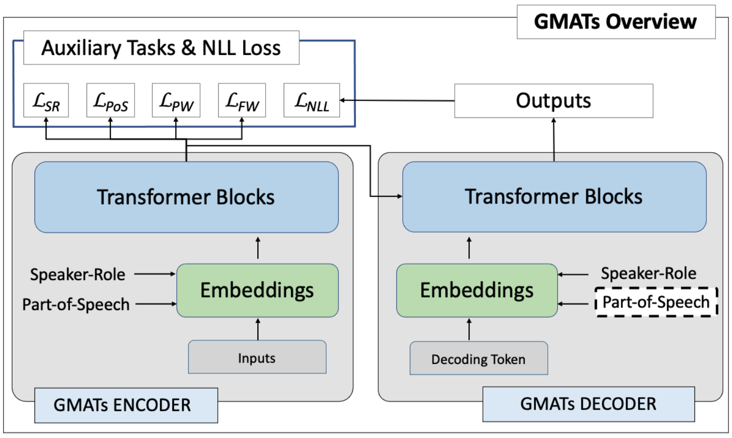


Figure 4.1: *Overview of the proposed GMATs model.*

structure based on the speaker participation?

In the subsequent sections, the proposed method will be demonstrated, followed by the reporting of experimental results conducted on two publicly available datasets.

4.1 Methodology

To incorporate context characteristics into the BART (Lewis et al., 2020), I devise a Generative Model with Auxiliary Tasks (GMATs) that comprises GMATs Encoder, a GMATs Decoder, and a GMATs Auxiliary Tasks module (see Figure 4.1). As shown in the figure, in the encoder, the speaker role information and part-of-speech information are incorporated, which enhances the model with dialogue characteristics. The same settings are also done for the decoder. To supervise the additional embeddings (i.e., speaker-role embedding and part-of-speech embedding), auxiliary tasks are used, training the model along with the NLL loss. The detailed explanation of the architecture is shown in Figure 4.2.

4.1.1 GMATs Encoder

The GMATs Encoder starts from raw texts, including all utterances from the context, i.e., $w_{n,1}, w_{n,2}, \ldots, w_{n,i}$ in Figure 4.2, where n means the n-th utterance and i denotes the i-th token in n-th utterance. All tokens in the utterances will be transformed into word representations ($h_{n,i}$). The GMATs Encoder inherits parameters from the BART Encoder except for the two new Embeddings: Speaker Role Embedding and Part-of-Speech Embedding.

Embeddings

In the GMATs, four embeddings are incorporated: position embedding, word embedding, speaker role embedding, and Part-of-Speech embedding.

Position embedding identifies token positions in a sequence. For the BART model, a pre-defined token position number is 1024, indicating 1024 as the maximum token number for an input sequence.

Word embedding covers all the tokens in the vocabulary by transferring the raw tokens to the embeddings. In this work, the two embeddings are inherited from the BART base model.[1]

Speaker role embedding cover two speakers, i.e., a poster and a responder, and therefore include two rows, identifying the two speakers.

Part-of-speech (PoS) embedding includes 4 elements: nouns, verbs, adjectives, and others.

Unlike word embedding and token position embedding, the latter two are initialized and trained from the start. Given a token t, its embeddings are obtained by looking up four embedding matrices. The outputs from the position embedding, word embedding, speaker role embedding, and part-of-speech embedding are denoted as e_{pe}, e_{we}, e_{sre}, and e_{pose}, respectively. The final token representation is then expressed as:

$$E_{n,i} = e_{pe} \oplus e_{we} \oplus e_{sre} \oplus e_{pose} \tag{4.1}$$

where i and n stand for the i-th token of the n-th utterance. $\oplus$ means element-wise plus.

[1] https://huggingface.co/facebook/bart-base

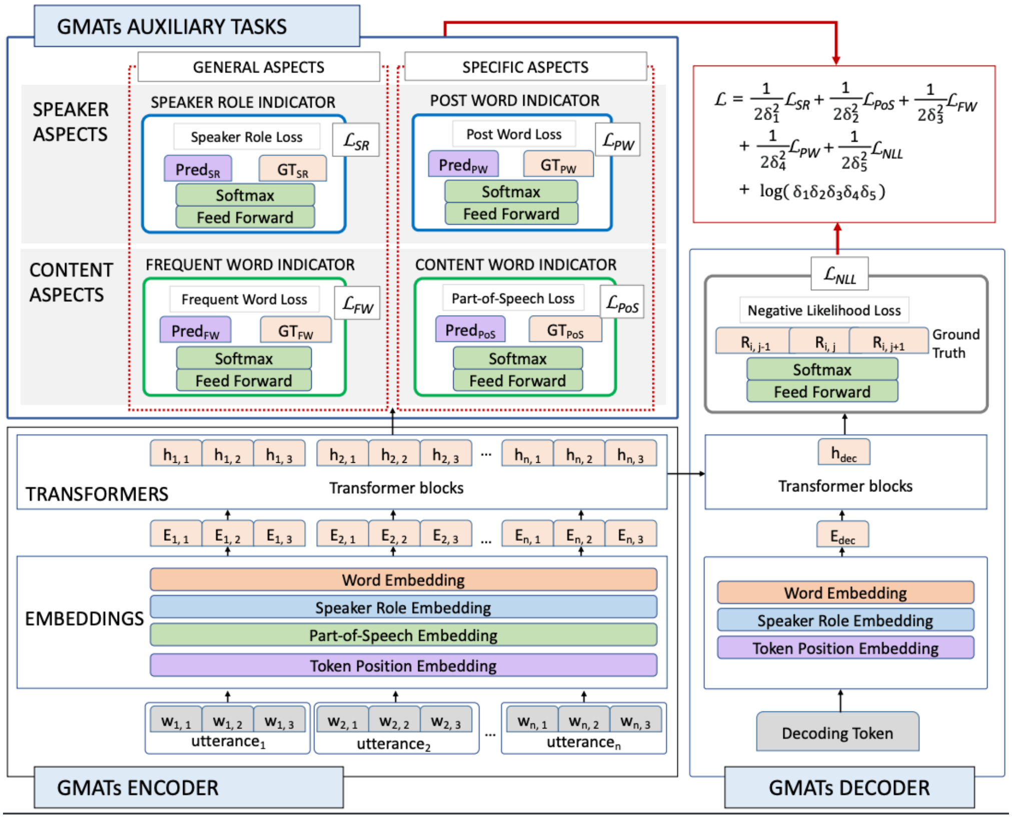


Figure 4.2: *Architecture of the proposed GMATs model. Speaker Role Embedding and Part-of-Speech Embedding are introduced to indicate the speakers' roles and part-of-speech. Four auxiliary tasks are designed to facilitate learning general and specific aspects of content and speakers' characteristics.*

Transformer Blocks

The Transformer Blocks of the BART model are employed in the GMATs model to process input. The Transformer Blocks transform each input word representation $E_{n,i}$ into an enhanced representation $h_{n,i}$ by applying Transformer layers. These enhanced representations capture the contextual information of the input words, considering both their positions and relationships with other words in the sequence. After processing the input through the Transformer Blocks, the resulting enhanced representations $h_{n,i}$ are fed into the decoder.

4.1.2 GMATs Decoder

Embeddings and Transformer Blocks. The architecture of the GMATs decoder is similar to that of the GMATs encoder. The additional part is the speaker role embedding. In the previous work (Bao et al., 2020), the speaker role embedding is added to the encoder's input but omitted in the decoding phase since it assumes that the first context utterance comes from the poster. However, the total number of utterances is variable across the dataset and there is no guarantee that, for example, the ground truth response actually comes from the responder.

Since the ground truth response to the responder role can be reliably attached, it is feasible to inject the responder role embedding into the decoder's embedding phase and enhance the representation of the decoding token. In terms of the part-of-speech characterization during decoding, the PoS information cannot be specified (since it is term vector that is processed within the model rather than the lexical term). Thus, the equation Eq. 4.2 used in the decoding phase includes:

$$E_{dec} = e_{pe} \oplus e_{we} \oplus e_{sre} \tag{4.2}$$

where E_{dec} is the decoding token's representation. Similar to the Encoder, the Transformer Blocks in the Decoder are exactly the same as the BART's Transformer Blocks and enhance the final token's representation h_{dec}.

Negative Likelihood Loss Function. Following the common practice of the language model, the h_{dec} will go through a feed-forward layer to fit the output requirements. A softmax operation is used to obtain the probabilities of each word in the vocabulary. The ground truth of the language model is the response, i.e., R. The Negative Likelihood Loss can be formalized as:

$$\mathcal{L}_{NLL} = -\sum_{i=1}^{N}\sum_{j=1}^{L} log\ p(R_{i,j}|C, R_{i,t<j}) \tag{4.3}$$

where, $R_{i,j}$ means j-th token in i-th response; t stands for time step; N denotes the total number of samples and L denotes the maximum response length. Therefore, given the context, C, and previously predicted tokens $R_{i,t<j}$, the objective is to predict the current token $R_{i,j}$.

4.1.3 Auxiliary Tasks

The auxiliary tasks are designed to fit four objectives: to discern the terms associated with speaker roles and the post, and the terms that belong to the set of frequent words from the training set and the content terms, i.e., having PoS with nouns, verbs, or adjectives. They are reflected in Figure 4.2 as the speaker role indicator, post word indicator, frequent word indicator, and content word indicator, respectively.

Each of them is treated as a classification task and use Binary Cross Entropy (BCE) for all tasks. As shown in Figure 4.2, each auxiliary task contains a feed-forward layer and a softmax operation to convert $h_{n,i}$ into the required outputs. (Note that different feed-forward layers in the four tasks do not share parameters.)

Furthermore, for the four auxiliary tasks, all words in the context are calculated and compared with the ground truth labels. For example, for all tokens in the context, $h_{n,1}, h_{n,2}, \ldots, h_{n,i}$, the predicted Speaker Role sequence, PoS sequence, Frequent Word sequence, and Post Word sequence are denoted as $Pred_{SR}$, $Pred_{PoS}$, $Pred_{FW}$, and $Pred_{PW}$, where $Pred_{SR}$, $Pred_{PoS}$, $Pred_{FW}$, and $Pred_{PW} \in \mathbb{R}^{1\times l}$ and l is the length of the input context.

Speaker Role Loss

As illustrated, there are only two speakers in the datasets, so it can be considered as a classification task. If an utterance comes from the poster, all words in the utterances are tagged as 0; otherwise 1 (refers to as the responder). Formally, the ground truth for Speaker Role Loss is constructed ($\mathcal{L}_{SR}$) as follows.

$$gt_{SR(n,\ i)} = \begin{cases} 1\text{, if word}_i\text{ comes from the responder} \\ 0\text{, if word}_i\text{ comes from the poster} \end{cases} \tag{4.4}$$

$$GT_{SR} = \{gt_{SR(1,\ 1)}, gt_{SR(1,\ 2)}, \ldots, gt_{SR(n,\ i)}, \ldots\} \tag{4.5}$$

The same as $Pred_{SR}$, $GT_{SR} \in \mathbb{R}^{1\times l}$. Then, the Speaker Role Loss function can be defined as:

$$\mathcal{L}_{SR} = BCE(Pred_{SR}, GT_{SR}) \tag{4.6}$$

Part-of-Speech Loss

Similar to the Speaker Role, the model predicts whether a word's PoS appears in the pre-defined PoS list (nouns, verbs, and adjectives).

$$gt_{PoS(n,\ i)} = \begin{cases} 1, \text{ if word}_i\text{'s PoS in PoS list} \\ 0, \text{ otherwise} \end{cases} \tag{4.7}$$

$$GT_{PoS} = \{gt_{PoS(1,\ 1)}, gt_{PoS(1,\ 2)}, \ldots, gt_{PoS(n,\ i)}, \ldots\} \tag{4.8}$$

Then, the Part-of-Speech Loss is formalized as follows.

$$\mathcal{L}_{PoS} = BCE(Pred_{PoS}, GT_{PoS}) \tag{4.9}$$

Frequent Word Loss

The frequent word is defined as any word that appears in the train set more than 800 times (following Khandelwal et al. (2018)), so the ground truth is:

$$gt_{FW(n,\ i)} = \begin{cases} 1, \text{ if word}_i\text{'s frequency >800} \\ 0, \text{ otherwise} \end{cases} \tag{4.10}$$

$$GT_{FW} = \{gt_{FW(1,\ 1)}, gt_{FW(1,\ 2)}, \ldots, gt_{FW(n,\ i)}, \ldots\} \tag{4.11}$$

Then, the Frequent Word Loss is formalized as follows.

$$\mathcal{L}_{FW} = BCE(Pred_{FW}, GT_{FW}) \tag{4.12}$$

Post Word Loss

A post is defined as the last utterance of the conversation. Thus, the binary ground truth can be defined as:

$$gt_{PW(n,\ i)} = \begin{cases} 1, \text{ if word}_i \text{ belongs to the post} \\ 0, \text{ otherwise} \end{cases} \tag{4.13}$$

$$GT_{PW} = \{gt_{PW(1,\ 1)}, gt_{PW(1,\ 2)}, \ldots, gt_{PW(n,\ i)}, \ldots\} \tag{4.14}$$

Similarly, Post Word Loss is formalized as:

$$\mathcal{L}_{PW} = BCE(Pred_{PW}, GT_{PW}) \tag{4.15}$$

4.1.4 Final Loss Function

Five loss functions have been introduced, including Eq. 4.3, Eq. 4.6, Eq. 4.9, Eq. 4.12, and Eq. 4.15 and I intend to use them to define the final, composite loss function used to optimize the GMATs model. Previous studies simply added up multiple loss functions Li et al. (2019b); Kim et al. (2020); Zheng et al. (2021) or defined a weighted sum by using manually set hyper-parameters Li et al. (2016b). Kendall et al. (2018) proposes to weigh different loss functions based on the homoscedastic uncertainty of each task. This Automatic Weighted Loss (AWL) method makes it possible to learn different weights for multiple loss functions. The AWL method is adopted and the final loss can be defined as follows:

$$\begin{aligned} \mathcal{L} = &\frac{1}{2\delta_1^2}\mathcal{L}_{SR} + \frac{1}{2\delta_2^2}\mathcal{L}_{PoS} + \frac{1}{2\delta_3^2}\mathcal{L}_{FW} \\ &+ \frac{1}{2\delta_4^2}\mathcal{L}_{PW} + \frac{1}{2\delta_5^2}\mathcal{L}_{NLL} + log(\delta_1\delta_2\delta_3\delta_4\delta_5) \end{aligned} \tag{4.16}$$

The objective is to minimize the final loss function by learning the parameters $\delta_1, \delta_2, \delta_3, \delta_4$, and δ_5 during the training process.

4.2 Experiment

4.2.1 Datasets and Metrics

In this work, two datasets are used: DailyDialog (Li et al., 2017b) and Wizard of Wikipedia (Dinan et al., 2019). The details of them are introduced in Sec. 2.3. These two datasets are selected because they contain various topics and are widely adopted in the previous works (Dinan et al., 2019; Kim et al., 2020). Most importantly, they are annotated by humans, regarding open-domain dialogues.

In terms of the metrics, as in previous research Ghazvininejad et al. (2018); Zhao et al. (2020a); Li et al. (2019b); Zheng et al. (2021), the dialogue generation models will be evaluated based on commonly adopted metrics: BLEU Papineni et al. (2002), Meteor Banerjee and Lavie (2005), Diversity Li et al. (2016a) and embedding-based metric, and BOW Embedding (Liu et al., 2016b). Please refer to Section 2.2 regarding the metrics' details.

4.2.2 Implementation Details

GMATs model is based on the BART model whose word embedding size is 768 and maximum input token length is 1024. The target token length is set to 64. For context, the latest 10 utterances are used. The learning rate is 5e-5. All of the experiments are trained on a single TITAN V GPU. The training phase is set with early-stopping criteria when it reaches a patient of 10 epochs. The time costs for DailyDialog, CMU-DoG, and WoW datasets are around 8, 14, and 16 hours.

4.2.3 Baseline Approaches

HRED: Serban et al. (2016) introduces a hierarchical architecture to get sentence-level and dialogue-level representations.

WSeq: Tian et al. (2017) considers using the post to assign similarities to the utterances so that the importance of utterances can be obtained during training.

HRAN: Xing et al. (2018) explicitly weighs each word and utterance for fine-grained representation aggregation.

ReCoSa: Zhang et al. (2019a) leverages the attention mechanism to update utterances' vectors so that all of the context utterances are considered attentively.

BART: Lewis et al. (2020) is a pre-trained model designed for generative tasks. The proposed model is built on top of BART.

PLATO: Bao et al. (2020) proposes to add the speaker role embedding to the pre-trained model, guiding the model with consideration of speaker information.

SI BART is Syntax-infused BART, (Sundararaman et al. (2021)) which uses PoS embedding, infusing syntax information into the pre-trained model BERT. I make an adaptation from the BERT to BART so that it is comparable to the proposed model.

4.3 Experimental Results and Analysis

The results of the DailyDialog and the WoW test-seen and test-unseen datasets are shown in Table 4.1.

4.3.1 Dialogue Generation Results

As shown in Table 4.1, in comparison with the non-pre-trained models HRED, WSeq, HRAN, and ReCoSa, the proposed GMATs model is consistently better in terms of BLEU, Meteor, Diversity, and Embedding-based metric scores. Being built on top of BART, it improves BART on all metrics by a large margin. PLATO introduces speaker role embedding to the pre-trained model. The results in Table 4.1 show that PLATO underperforms on BLEU, Meteor, and Embedding-based scores but achieves the best diversity performance on

Table 4.1: Automatic evaluation results on DailyDialog and Wizard of Wikipedia datasets. * means significant test value with $p < 0.05$, in comparison with the proposed GMATs . ‘w/o’ means without a certain loss function for the ablation study. All values are expressed as percentages (%).

Models	DailyDialog					
	BLEU-2	Meteor	Div-2	Average	Extrema	Greedy
HRED (Serban et al., 2016)	5.88*	6.67*	1.08*	67.34*	40.93*	39.03
WSeq (Tian et al., 2017)	6.28*	6.70*	4.45*	66.53*	40.77*	36.81*
HRAN (Xing et al., 2018)	6.40*	6.64*	6.05*	66.17*	40.24*	35.82*
ReCoSa (Zhang et al., 2019a)	6.11*	6.45*	1.95*	66.15*	41.31*	36.66*
BART (Lewis et al., 2020)	8.07*	7.55*	33.76*	66.54*	40.99*	36.07*
PLATO (Bao et al., 2020)	7.18*	7.10*	**49.01***	62.32*	37.64*	33.07*
SI BART (Sundararaman et al., 2021)	12.46*	10.25*	37.81*	69.24*	43.17*	38.59*
GMATs GMATs	**17.10**	11.72	45.74	70.89	44.82	39.77
w/o $\mathcal{L}_{FW}$	16.96	**11.75**	45.81	**71.19**	**44.95**	**40.07**
w/o $\mathcal{L}_{SR}$	16.09*	11.20*	48.99*	69.79*	43.97*	39.29*
w/o $\mathcal{L}_{PW}$	15.45*	10.93*	47.28*	69.74*	44.00*	39.07*
w/o $\mathcal{L}_{PoS}$	15.03*	10.79*	47.31*	69.55*	43.74*	39.22*
Models	Wizard of Wikipedia test seen					
	BLEU-2	Meteor	Div-2	Average	Extrema	Greedy
HRED	6.65*	6.05*	10.46*	61.42*	29.53*	36.63*
WSeq	6.94*	6.44*	9.15*	63.18*	31.26*	37.70*
HRAN	8.16*	7.12*	12.38*	64.82*	33.22*	39.28*
ReCoSa	6.80*	6.38*	9.93*	63.18*	31.24*	37.92*
BART	10.61*	9.12*	39.34*	69.14*	41.46*	41.14*
PLATO	9.80*	9.15*	35.00*	68.53*	41.45	40.11*
SI BART	11.13*	9.56*	**40.30***	69.07*	40.74*	40.86*
GMATs GMATs	**12.55**	10.21	37.97	69.97	41.15	41.31
w/o $\mathcal{L}_{FW}$	12.52	**10.27**	37.10	**70.19**	**41.47**	41.42
w/o $\mathcal{L}_{SR}$	12.41*	10.22	36.93	69.69	41.33	41.37
w/o $\mathcal{L}_{PW}$	12.41*	10.15*	37.62	69.98	41.16	41.22
w/o $\mathcal{L}_{PoS}$	12.46*	10.25	37.81	70.16	41.53*	**41.60***
Models	Wizard of Wikipedia test unseen					
	BLEU-2	Meteor	Div-2	Average	Extrema	Greedy
HRED	5.77*	5.63*	9.07*	60.29*	27.59*	35.91*
WSeq	6.22*	6.01*	7.35*	62.02*	28.88*	36.92*
HRAN	6.54*	6.13*	10.35*	62.86*	30.16*	37.70*
ReCoSa	6.16*	5.93*	8.04*	62.05*	29.06*	37.26*
BART	9.01*	8.15*	28.99*	66.50*	36.04*	38.18*
PLATO	8.73*	8.50*	25.32*	67.13*	**38.22***	38.75*
SI BART	9.40*	8.57*	**30.72***	67.58	37.36	39.29
GMATs GMATs	**9.65**	8.69	24.62	67.49	37.72	39.07
w/o $\mathcal{L}_{FW}$	**9.65**	8.62	24.50*	67.10*	37.57	39.05
w/o $\mathcal{L}_{SR}$	9.57*	**8.91***	22.53*	**68.01***	37.81	**39.44***
w/o $\mathcal{L}_{PW}$	9.56*	8.58*	25.37*	66.85*	37.49	38.89
w/o $\mathcal{L}_{PoS}$	9.42*	8.57*	24.46*	67.01*	37.33*	38.91

DailyDialog. Similarly, SI BART incorporates PoS embedding and generally performs worse than GMATs . However, it achieves higher scores than other baselines on most measurements confirming that PoS does help the generative models.

Generally, the GMATs model outperforms most of the baseline approaches on div-1 and div-2 scores except for PLATO and SI BART on DailyDialog and WoW datasets respectively.

4.3.2 Effectiveness of Each Component

Ablation Study. In Table 4.1, the results of the ablation study are reported to consider the contribution of individual auxiliary tasks to the model. The comparison groups include:

(1) w/o $\mathcal{L}_{FW}$: remove Frequent Word loss function;

(2) w/o $\mathcal{L}_{SR}$: remove Speaker Role loss function;

(3) w/o $\mathcal{L}_{PW}$: remove Post Word loss function;

(4) w/o $\mathcal{L}_{PoS}$: remove part-of-speech loss function.

By considering BLEU-2 scores, similar trends can be observed in those two datasets. By removing post word loss ($\mathcal{L}_{PW}$), speaker role loss ($\mathcal{L}_{SR}$), and part-of-speech loss ($\mathcal{L}_{PoS}$), the performance decreases significantly. Especially on the DailyDialog dataset, BLEU-2 drops by around 9.7% (from 17.10% to 15.45%), 6.0% (from 17.10% to 16.09%), and 12.1% (from 17.10% to 15.03%). That demonstrates the effectiveness of adding these three context characteristics (i.e., post word, speaker role, and Part-of-Speech). On the other hand, removing frequent words and its loss function $\mathcal{L}_{FW}$ does not have a significant effect on BLUE-2. However, deleting $\mathcal{L}_{FW}$ improves the Embedding-based metrics significantly: the average score from 70.89% to 71.19% (DailyDialog, with a similar trend on the WoW dataset). I attribute this to the fixed word frequency threshold of 800 (Khandelwal et al., 2018). It is likely that for different datasets, the effective threshold needs to adapt accordingly. By considering GMATs general and specific auxiliary tasks related to the speaker roles and content characterization (Figure 4.2), it can be confirmed that more *specific aspects* provide a more significant contribution to the performance according to BLEU-2. Removing PoS (specific factor) leads to a higher reduction (score 15.03) than the removal of FW (general factor) (16.96). Similarly, the use of the speaker role indicator vs. post word indicator shows that removing post indicator PW causes a high drop (to 15.45) compared to removing *general* speaker role SR (drop to 16.09). Overall, the combination of all aspects achieves the best performance.

4.4 Summary

Context-aware dialogue generation has benefited from novel techniques that use pre-trained models as a basis for further enhancements. In particular, the use of Auxiliary Tasks has opened up opportunities for introducing new aspects of the dialogue context. However, the principle of a systematic selection of context features is still an open question. In this work, Auxiliary Tasks which correspond to the two content characteristics and two speaker characteristics are devised. Each pair deals with more *general aspects* (frequent words and speaker roles) compared to more *specific aspects* (content words and post words indicator). Through ablation study with DailyDialog and Wizard of Wikipedia datasets, we can know that more *specific aspects* have a more significant contribution to the BLEU-2 and Meteor scores.

To generate dialogue, background knowledge is provided in addition to contextual information. Research has shown that knowledge is more informative than context alone, and incorporating knowledge improves the performance of generative models, as indicated by studies such as Ghazvininejad et al. (2018); Kim et al. (2020). Therefore, selecting the appropriate knowledge is an important area of research. To evaluate knowledge-grounded generative models benchmarks such as Wizard of Wikipedia and CMU-DoG have been developed. Wikipedia articles are typically used as a source of background knowledge, normally consisting of multiple sentences. The task of knowledge selection involves choosing the most relevant knowledge from the available set. This issue will be discussed in Chapter 5 as future work.

Chapter 5

Enhancing Conversational Dialogue Models with Grounded Knowledge

In comparison to generative models that only consider the context, knowledge-grounded models are capable of generating more informative and engaging responses. Previous research (Shao et al., 2017; Li et al., 2016a) has shown that conversational models often produce uninteresting and uninformative answers. Humans answer questions based on their own knowledge, and it is impossible to answer a question correctly without sufficient knowledge. Similarly, the information necessary for generating good conversational responses is often stored in external knowledge sources. As evidence for this argument, Table 5.1 presents examples of real-world online conversations from Reddit. The words highlighted in red demonstrate that background knowledge often contains information or vocabulary that is essential for generating a reasonable response, but that may not be present in the post. It can also be observed that a knowledge-free model, such as the Vanilla Transformer, struggles to generate a proper response compared to a knowledge-grounded solution, such as the proposed model (TED).

Recent research (Yang et al., 2018; Tian et al., 2017; Madotto et al., 2018a;

Table 5.1: The red-tagged vocabulary in targets (responses) appears in knowledge, but not in the post. Incorporating the appropriate vocabulary from the knowledge, the proposed knowledge-grounded model (TED) is able to generate superior responses than the model (Vanilla Transformer) that disregards knowledge.

Post: Do you know anything about the narcissus plant? **Response:** I do know that it is pretty much a spring perennial plant. **Knowledge:** A perennial plant or simply perennial is a plant that lives for more than two years. **Vanilla Transformer:** Controlled love n't that the is a much better lot and plant. **TED:** Deer do know that it is, much of weird perennial plant.
Post: Do you know when the Mustang was first made? **Response:** Yeah the original ford Mustang was manufactured in 1962! It was a two seater concept car. **Knowledge:** the ford Mustang is an American car manufactured by ford. **Vanilla Transformer:** Controlled, term formula Mustang was created by 1962. They was founded compact seater concept car. **TED:** Deer the modern civic Mustang was manufactured in 1962! It was a British seater concept car.

Ghazvininejad et al., 2018) has demonstrated that incorporating relevant background knowledge can lead to better and more varied responses. Tian et al. (2017) develops a hierarchical model with two RNN layers to capture contextual information and found that the hierarchical approach outperformed the non-hierarchical one (RNN-based models). Ghazvininejad et al. (2018) proposes a fully data-driven model that employs Memory Neural Networks and multi-task learning to incorporate knowledge (MemNN-based models), leading to more diverse responses in both Twitter and Foursquare datasets. Dehghani et al. (2019a) uses Transformer and Universal Transformer models to apply multi-hop reasoning to background knowledge, utilizing knowledge from less prominent documents in the retrieval process (Transformer-based models).

While RNN-based, MemNN-based, and Transformer-based models have been shown to effectively incorporate knowledge in various contexts, at the time of publishing this work, there has not been a comprehensive evaluation that compares these approaches. Additionally, it is still unclear how to optimally select external knowledge.

In this chapter, the objective is to bridge the gap and answer the following research questions (RQs):

1. **How do different types of knowledge-grounded generative mod-**

els perform? Typical approaches from all three types are selected and systematic experiments are conducted to assess their effectiveness in producing high-quality and diverse responses.

2. **How does the amount of knowledge affect the generative models' performance?** For different models, it is not clear how much knowledge (e.g., top 3, 10, or 20 sentences) should be retrieved to optimally enhance the conversational model, trading off between the relevant knowledge and noise.

3. **Can we find an effective approach that optimally selects appropriate information from the utterance and the external knowledge?** By assuming that the utterance and the external knowledge contribute to the response in different ways, a novel Transformer with Expanded Decoder (TED) model is proposed.

In TED, two additional functional modules, an attention-weighting layer, and an attention-merging layer are introduced based on the Transformer decoder architecture. Optimally tuning the weights to balance various sources of evidence, TED consistently outperforms previous models on two data sets, in terms of both quality and diversity.

5.1 Fundamental Models

To be self-contained, a brief introduction to the fundamental models used in this work is presented, encompassing RNN-based, MemNN-based, and Transformer-based models. Representative models are chosen within each category to serve as the baseline approaches.

5.1.1 RNN-based Sequence-to-Sequence Models

In the generative model community, the Seq2Seq model (Bahdanau et al., 2014) is widely adopted as a baseline. Feeding a context C into the Seq2Seq model, the encoder encodes it to the vector representation (by the last hidden state of an RNN), and the decoder is used to interpret the vector to the target response R. Normally, the encoder and decoder are RNNs with LSTM cells or GRU cells.

Retrieve and refine (Weston et al., 2018) and *Pointer generator* (Yang et al., 2018) are two Seq2Seq variant models. The former retrieves external knowledge and extends the context with top-ranked words of the retrieval for refining context. The latter model introduces a copy mechanism into the Seq2Seq model. The output words' probabilities come from both the generative model and the context's words. The copy-mechanism, theoretically, can improve the diversity of the responses because the final word probabilities come from two sources and thus the **Pointer Generator** model is chosen as one of the baseline models. Even though LSTM or GRU can gain long-term memory, it is still difficult to tackle a very long sequence. *Hierarchical Networks* (Serban et al., 2016) adopt a traditional hierarchical recurrent encoder-decoder architecture (HRED) by combining with pre-trained embedding. It builds an utterance-level RNN layer on top of a term-level layer, by which the model can gain an interrelation between the knowledge and the context. The limitation of this approach lies in its inefficiency because it poses more computation in the model. The study Tian et al. (2017) proposes a Weight Sequences (WSeq) model and proclaims that their proposed hierarchical models perform better than other context-injecting methods, and thus the proposed model will be compared to the **WSeq-Sum** and **WSeq-Concat**.

5.1.2 End-to-End Memory Network

Memory neural network (MemNN) is first proposed in Weston et al. (2015). Sukhbaatar et al. (2015) proposes a more practical version, the End-to-End Memory Networks (E2E MemNN). To formally illustrate the E2E MemNN, for i-th data sample, C_i is denoted as the query, and $K = \{k_i\}$ (i.e., a set of knowledge sentences is given to the context C_i) as a knowledge set. Here, knowledge is to be stored in memory. The knowledge would be changed to d-dimensional memory vectors $\{m_i\}$ by embedding k_i with a trainable embedding matrix A($d \times V$, where V is the vocabulary size) (Sukhbaatar et al., 2015). The same process will also be done to the query C_i, but with a different embedding matrix B (B has the same dimension as A), and then C_i will be converted to an internal state u_i. Following Ghazvininejad et al. (2018), t_i and f_i^j are used to represent the bag of words of C_i and k_i (where j means the j-th word), and thus the E2E MemNN can be formulated as follows.

$$u_i = Bt_i \tag{5.1}$$

$$m_i^j = Af_i^j \tag{5.2}$$

$$c_i^j = Cf_i^j \tag{5.3}$$

$$p_i^j = \text{softmax}(u_i^T m_i^j) \tag{5.4}$$

$$o_i = \sum_{j=1} p_i^j c_i^j \tag{5.5}$$

$$\hat{u}_i = o_i + u_i \tag{5.6}$$

where, $A, C \in \mathbb{R}^{d \times V}$ are two embedding matrices and should be trained in the E2E MemNN. Originally, A and C were different matrices: A is the input embedding matrix for C_i while C is the output embedding matrix. I follow the "Adjacent" method shown in Sukhbaatar et al. (2015) where $A^{k+1} = C^k$ and $B = A^1$ (k is the k-th layer). p_i is a softmax similarity based on Sukhbaatar et al. (2015), and it is used to choose which part in the memory is most relevant to the query sequence. Then o_i is employed to summarize the potentially useful content in the memory, and finally, it is added to the query vector u_i. For more details about E2E MemNN, please referring to the original paper Sukhbaatar et al. (2015).

In comparison with the original E2E MemNN model, it is not used to predict the target directly. Instead, the E2E MemNN is taken as the encoder to generate the hidden states and then input it to an RNN decoder with the GRU cell. The knowledge is infused one by one into the query, which is memory-friendly and easy to implement.

On the other hand, E2E MemNN can reason between the knowledge and the query, and to some extent handle long-term memory because it chooses memory by the semantic similarity between the query and knowledge. *Memory-to-sequence* (Madotto et al., 2018b) incorporates the pointer generator into an *E2E MemNN*, potentially improving the memory network's performance. However, both of them fail to effectively distinguish noise from truly useful information. Multi-task learning (Ghazvininejad et al., 2018; Luan et al., 2017) takes E2E MemNN as a basic backbone to jointly train multiple tasks. When different tasks converge together, the shared parameters in the decoder can be affected implicitly by the model that injects knowledge. Like transfer learning, the three models can learn from each other. Because of these potential advantages, this typical MemNN-related work is adopted in this work. It is also the baseline model of the DSTC-7 task.

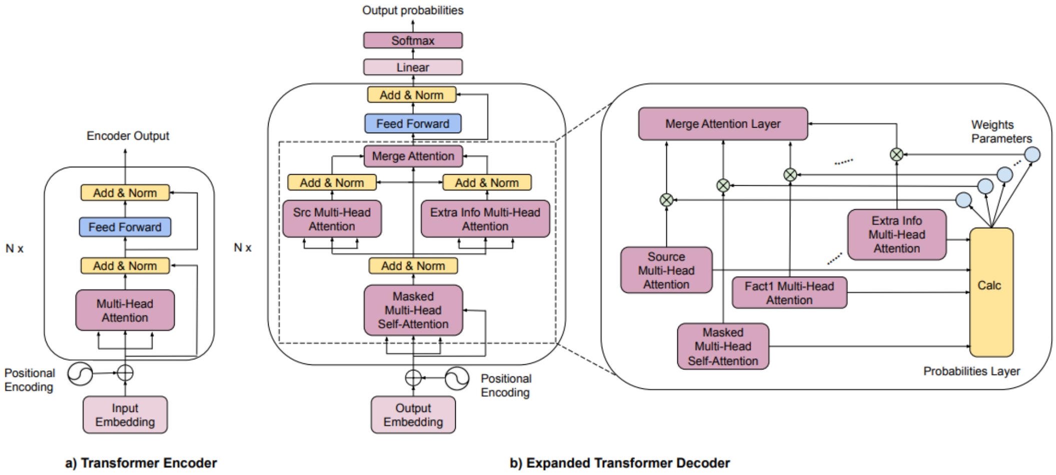
5.1.3 Transformer-based Models

Section 2.1.2 demonstrates that the attention mechanism is the critical component in the Transformer. This mechanism allows transformer-based models to reason within a sequence by calculating the attention of each word with every other word in the sequence. The Transformer quickly became the leading model for Sequence-to-Sequence tasks after its introduction. However, the vanilla Transformer has a limitation in that it cannot incorporate external knowledge. The *Generative Transformer Memory Networks* (Dinan et al., 2019) overcome this limitation by using two Transformer units to select the best knowledge and generate a response simultaneously, although it can only incorporate one unit of external knowledge. *SDNet* (Zhu et al., 2018) suggests containing all of the history hidden states to achieve competitive performance. However, this model is memory-intensive. The *Multi-hop Transformer* (Dehghani et al., 2019b) introduces a multi-hop reasoning layer based on the Universal Transformer that can incorporate multiple pieces of knowledge from the document level rather than just the token level. As a result, it is suitable for factoid tasks such as SearchQA and Quasar-T but not open chitchat tasks.

5.1.4 Discussion on These Methods

RNN-based, End-to-End Memory Network-based and Transformer-based generative models have their own characteristics and advantages. RNNs rely on sequential processing, allowing them to capture temporal dependencies in dialogues effectively, but they may struggle with capturing long-range dependencies and often suffer from the vanishing gradient problem. End-to-end memory Networks, on the other hand, explicitly store and retrieve information from a memory matrix, enabling them to handle more extensive contexts but may be less efficient for short-term dependencies. Transformer-based models have gained prominence due to their self-attention mechanism, excelling at capturing both short and long-range dependencies, making them highly effective for dialogue generation. A commonality across these models is their capacity to generate coherent and contextually relevant text, but the key difference lies in the mechanisms they employ for handling contextual information. As well-known recently, the Transformer-based models, such as BERT and GPTs, are particularly versatile in handling various NLP tasks and achieving state-of-the-art performance.

Prior to the release of pre-trained models, these three categories of generative models held prominence in dialogue generation tasks. In this work, the distinction between generative models that utilize at least one of these three backbones is examined. Leveraging their advantages and disadvantages, a novel architecture called Transformer with Expanded Decoder (TED) is im-



a) Transformer Encoder

b) Expanded Transformer Decoder

Figure 5.1: *The TED model: a) a Transformer encoder (the same as the vanilla Transformer); b) an expanded decoder. The right-hand side sub-figure is the detailed inner modules. For simplification, some modules are omitted in the right-hand side sub-figure. In b), the 'Extra Info' stands for external knowledge, and the 'Calc' means the probabilities calculating layer.*

plemented to incorporate multiple knowledge units for response generation.

5.2 Methodology

The proposed Transformer with Expanded Decoder (TED) model is an extension of the Transformer architecture that enhances its ability to automatically adjust attention weights based on both the context and external knowledge. While TED's encoder remains unchanged from the vanilla Transformer, the decoder module has been expanded to meet the requirements. Figure 5.1a) illustrates the vanilla Transformer encoder, while Figure 5.1b) depicts the TED's decoder. The external knowledge, which contains all of the necessary information, is denoted by 'Extra Info'. Two functional layers, the 'Probabilities Layer (PL)' and 'Merge Attention Layer (MAL)', are employed to incorporate this extra information. The PL module is responsible for generating 'Weights Parameters' that automatically learn weights for each additional information, and these parameters are subsequently input into the MAL module to be merged. Further details regarding the formulations are provided below.

Word-Level Attention As shown in Figure 5.1 a), the encoder part is the same as the vanilla Transformer. I conduct a word-level Transformer process for the context C and knowledge K. For each word in the sequence, the trainable embeddings is used rather than a certain pre-trained embedding models (e.g., Zhu et al. (2018) uses the GloVe proposed by Pennington et al. (2014)).

Inner-Attentions In the decoder component (Figure 5.1 b), multi-head attentions are integrated between the decoder inputs and both the context and knowledge. This is based on the assumption that context and knowledge impact the generated word in distinct ways. The mathematical equations for these multi-head mutual attentions are presented below.

$$v_{C\_dec} = Attention(v_{dec}, v_C, v_C) \tag{5.7}$$

$$v_{f\_dec} = Attention(v_{dec}, v_k, v_k) \tag{5.8}$$

where v_{C_dec} and v_{k_dec} means multi-head attentions between C, k and the decoder inputs respectively. v_C, v_k and v_{dec} are their self-attention representations.

Expanded Transformer Decoder The Transformer decoder, originally, has a self-attention layer, a multi-head mutual attention layer, and a feed-forward layer. Here in the TED, two extra functional layers are designed: PL and MAL, as shown in the right-hand side of Figure 5.1 b). After getting inner-attentions, v_{C_dec} and v_{k_dec}, they are inputted into the PL to get the weights parameters P which will be multiplied with v_{C_dec}, v_{k_dec} and v_{dec} itself. These weights are trainable parameters in the PL.

$$P = W_C * v_{C\_dec}^T + \sum W_{k_j} * v_{k_j\_dec}^T + W_{dec} * v_{dec}^T + b \tag{5.9}$$

where W_C, W_{k_j}, W_{dec}, and b are trainable parameters. It is flexible to inject several pieces of knowledge because the weight parameters here are calculated automatically. The MAL merges differently weighted attention scores as follows:

$$v_{merge} = p_C * v_{C\_dec} + \sum p_{k_j} * v_{k_j\_dec} + p_{dec} * v_{dec} \tag{5.10}$$

Here, p_C, p_{k_j} and p_{dec} come from the Equation 5.9. The MAL is just a functional layer for merging different attentions together.

5.3 Experiment

5.3.1 Datasets

I use two data sets for the experiments: Reddit and Wizard of Wikipedia (introduced in Sec. 2.3). Both of them contain a set of contexts, responses, and background knowledge. Given the fact that the background knowledge provided in the original Reddit dataset is very noisy, a filtered Reddit dataset is obtained, in which the provided background knowledge contains useful information.

5.3.2 Implementation Details

Data pre-processing. For all the datasets used in this study, which include the Reddit and WoW datasets, a consistent set of pre-processing steps is applied. The samples are filtered based on the response sequence length, which is constrained to fall within the range of 8 to 30 terms. Following the approach outlined by Ghazvininejad et al. (2018), a vocabulary of 50k terms is selected based on term frequency ranking. This vocabulary is shared between both the contexts and the responses, and the maximum response length is capped at 30 terms for both datasets. To cover more than 80% of the sequence lengths of the contexts and knowledge sentences, a statistical analysis of the length distributions of the contexts and knowledge sentences is conducted separately for the Filtered Reddit and WoW datasets. Consequently, the maximum context and knowledge sentence length is set to 100 for the Filtered Reddit dataset, while lengths of 30 are assigned to the contexts and knowledge sentences in the WoW dataset, respectively.

Knowledge pre-processing. To incorporate potentially relevant knowledge into the model, distinct retrieval techniques are employed during both the training and testing stages. During training, the approach involves the use of an "oracle retrieved knowledge set." This process entails selecting "useful words" (as elaborated in Section 2.4) as queries for each context. These queries are then employed to retrieve all pertinent knowledge sentences. Subsequently, these sentences are ranked based on the number of useful words they contain, with the highest-ranked sentences being more likely to contain valuable information for model training. During testing, a straightforward TF-IDF weighting scheme is utilized to retrieve relevant knowledge sentences. This method involves ranking the sentences based on their similarity score to the context.

Setup for models. Regarding the models, the proposed approach, in accordance with Ghazvininejad et al. (2018), entails constructing a gated Seq2Seq model that consists of three layers of GRUs. Each layer has a hidden state dimension of 100, and both the encoder and decoder share this state dimension. Furthermore, the word embedding dimension is set to 100, and the Adam optimizer with a learning rate of 0.001 is utilized.

For the E2E MemNN model, a memory embedding size of 100 is allocated, and the hop step is set to be 3. In the case of the Transformer models, default settings are employed[1]. The Transformer encoder and decoder stacks are configured with 3 layers, and multi-head attention employs 4 heads and 100 attention dimensions.

Computational costs. Based on the experimental setup, all of the experiments are conducted on a single GPU (GeForce GTX 1080). For the Original Reddit data set, the TED takes 2 days for training, while for the Filtered Reddit data set and the Wizard of Wikipedia dataset, it takes around 3 hours for training. When predicting, if an end token is predicted or the maximum

[1] https://github.com/kpot/keras-transformer

length is reached, the process terminates.

Filtered Reddit Dataset It can be observed that external links on Reddit often contain a substantial amount of noise, and many of the external knowledge articles lack useful information. To quantify the extent of noise, a metric named the "useful-words-rate" (UWR) is introduced. The calculation involves the removal of stop words and the identification of words that appear in both the response and the corresponding external knowledge article but not in the context; these words are deemed "useful." The UWR is defined as the frequency of these useful words divided by the total number of words in the context.

Empirical data indicates that the Wizard of Wikipedia dataset exhibits a UWR of 193%, which is significantly higher than the Original Reddit dataset's UWR of 46%. To focus the experimentation on datasets that contain useful information in the background knowledge articles, a Filtered Reddit dataset is created. This dataset consists of only the top Reddit samples selected based on the UWR metric from the Original Reddit dataset, resulting in 91,344 samples for the training set and 5,000 for the test set.

5.3.3 Metrics

As described in Section 2.2, my primary measures for evaluating the relevance between the predicted and ground truth responses are BLEU-2 and METEOR. BLEU-2 measures the co-occurrence of bi-gram terms in the generated response and the reference response, while METEOR takes into account the presence of synonyms and common word stems to better capture semantic similarity. Additionally, Div-2 is used to gauge the diversity of the generated responses. Div-2 is calculated by dividing the number of distinct bi-grams in the generated responses by the total number of generated bi-grams.

5.3.4 Baseline Approaches

Seq2Seq-Attn: Bahdanau et al. (2014) introduces an attention mechanism to the sequence generation by looking back to the input sequence with attention scores.

Pointer-Generator: See et al. (2017) considers copy mechanism to the summarization task. The predicted sequence can either be generated by the model or be copied from the input texts.

WSeq-Sum and WSeq-Concat: Tian et al. (2017) explicitly weighs context utterances by the similarity scores between the post and context utterances.

Multi-Task: Ghazvininejad et al. (2018) leverage the multi-task learning paradigm to train three tasks in the meantime, including two response generation tasks and a sequence prediction task.

Transformer: Vaswani et al. (2017) make full use of the attention mechanism to update the word representation. Each word will attend to other words' representation, which makes the word representation semantically abundant.

5.4 Experimental Results and Analysis

Table 5.2: For the Filtered Reddit and Wizard of Wikipedia data sets: Compare different types of knowledge-grounded models on automatic metrics. TED is used as the base model to do the significant test. * stands for significant test value $p < 0.05$.

Data sets	Filtered Reddit data set			Wizard of Wikipedia data set		
Metrics	BLEU-2(%)	Meteor(%)	Div-2(%)	BLEU-2(%)	Meteor(%)	Div-2(%)
Seq2Seq-Attn	2.06*	3.70*	0.17*	4.41*	4.60*	0.89*
Pointer-Generator	3.55*	2.80*	4.00*	7.08*	6.80*	8.30*
WSeq-Sum	4.39*	5.39*	1.43*	7.69*	6.98*	12.96*
WSeq-Concat	4.20*	5.15*	1.75*	7.07*	6.47*	12.70*
Multi-Task	4.16*	5.04*	0.36*	6.78*	6.26*	8.00*
Transformer	3.04*	4.15*	0.06*	7.15*	6.60*	7.30*
TED	**5.01**	**5.60**	**9.40**	**9.47**	**8.45**	**16.20**

5.4.1 Results and Analysis of the Models

The performance of all models on both the Filtered Reddit and the Wizard of Wikipedia datasets can be seen in Table 5.2. These results reflect the performance of the proposed model in the scenario where the background knowledge is less noisy and contains useful information. Looking at Table 5.2, there are two groups to discuss: (1) RNN-based models vs. MemNN-based models; (2) Transformer-based models vs. Other models. RNN-based models mainly include the Pointer-Generator model and the hierarchical RNN model. The 'Multi-Task' is built based on the E2E MemNN model. Moreover, the Seq2Seq-Attn model, the Pointer-Generator model, and the Transformer model are three models that do not infuse any knowledge.

RNN-based models vs. MemNN-based models Table 5.2 reveals two consistent patterns. Firstly, the knowledge-injecting models generally perform better, as indicated by the higher BLEU-2 and Meteor scores, with only the Pointer-Generator model outperforming the Multi-Task model on the Wizard of Wikipedia dataset. This suggests that incorporating external knowledge into the context can enhance performance, which is consistent with previous research Weston et al. (2018); Tian et al. (2017); Ghazvininejad et al. (2018); Luan et al. (2017). Secondly, there are differences in diversity among the models across the two datasets. While the WSeq models and the multi-task model exhibit relatively better results on the Wizard of Wikipedia dataset, they fare worse than the Pointer-Generator model on the Filtered Reddit dataset. This indicates that the knowledge-infusing models can be sensitive to the dataset, with the human-generated Wizard of Wikipedia dataset being of higher quality than the Filtered Reddit dataset.

Among the knowledge-infusing models, the two WSeq models perform similarly on both datasets and generally outperform the other RNN-based and MemNN-based models. While the Multi-Task model does not surpass the WSeq models, it exhibits better performance than the models without knowledge infusion, which can be attributed to its parameter-sharing mechanism. Like transfer learning, when the three tasks are trained together, each task learns parameters from itself and the other two models, enabling it to acquire additional information from the other tasks.

Transformer-based models vs. Other models On both the Filtered Reddit and WoW data set, compared to other non-knowledge-injecting models, the Transformer model outperforms the Seq2Seq-Attn model but performs worse than the Pointer-Generator model. With regard to diversity scores on the Filtered Reddit dataset, the Transformer model exhibits the worst performance. This can be attributed to two primary factors. Firstly, it lacks explicit conditioning on knowledge, which distinguishes it from other models. While the alternative models either directly condition their responses on injected knowledge (such as WSeq and Multi-Task) or employ a copy mechanism to enhance variability, the Transformer relies solely on previous tokens and its attention mechanism to generate text. Secondly, the impact of the training dataset is substantial. Within the Reddit dataset, the responses are extracted from a forum where numerous users frequently provide low-quality content, such as the commonly encountered phrase "I am not sure about it." Consequently, Transformers tend to replicate the prevalent patterns found in the data, often resulting in the generation of generic responses.

While hierarchical RNN models demonstrate competitive capabilities, they are constrained in terms of extensibility and untapped potential. Given my understanding of the potential reasons behind the vanilla Transformer's inability to surpass other baseline methods, I delve deeper into its possibilities by introducing knowledge conditioning on top of the Transformer, i.e., the proposed TED model. These findings reveal significant improvement across all metrics for both datasets. As demonstrated by the anecdotal examples in Table 5.1, the TED model effectively utilizes background knowledge to generate informative responses. Significant tests using a two-tailed student t-test is conducted on all metrics, and the results indicate that TED is statistically superior to the baselines in both quality and diversity.

5.4.2 Results and Analysis of The Right Amount of Knowledge Sentences

Table 5.3: Injecting a different number of knowledge sentences to the TED on the Wizard of Wikipedia dataset. The TED gets a peak performance at 3, so the 3-knowledge-sentence model is taken as the base model to do the significant test. 'KS' means knowledge sentence. * stands for significant test value $p < 0.05$.

Method	TED		
Metrics	BLEU-2	Meteor	Div-2
1 KS	9.47	**8.45***	16.20*
2 KS	8.94*	7.73*	15.20*
3 KS	**9.62**	8.04	**22.20**
4 KS	8.96*	7.71*	18.00*
5 KS	9.09*	7.71*	18.76*
10 KS	8.88*	7.78	16.30*
11 KS	8.94*	7.82	17.40*
12 KS	8.95*	7.70*	16.00*
13 KS	8.88*	7.70*	13.70*
14 KS	8.98*	7.75*	14.70*
15 KS	9.13	7.80	14.40*

Table 5.4: Injecting more external knowledge to the WSeq-Sum model on the Wizard of Wikipedia dataset. The WSeq-Sum model gets a peak performance at 12 KS, so the 12 KS model is taken as the base model to do the significant test. 'KS' means knowledge sentences. * stands for significant test value $p < 0.05$.

Method	WSeq-Sum		
Metrics	BLEU-2	Meteor	Div-2
1 KS	7.69*	6.98*	12.96
2 KS	7.90*	7.00*	15.80
3 KS	7.68*	6.90*	16.75*
4 KS	7.87*	7.07*	16.75*
5 KS	7.96*	7.20*	17.59*
10 KS	8.29	7.44	19.48
11 KS	8.10*	7.27*	18.71
12 KS	**8.39**	**7.49**	19.24
13 KS	7.91*	7.31	19.46
14 KS	8.23	7.46	**19.96***
15 KS	8.16	7.35	18.96

The top 2 performing models (TED and WSeq-Sum) on the Wizard of Wikipedia dataset are selected from Table 5.2 and analyze the optimal amount of external knowledge to infuse by experimenting with injecting different numbers of top retrieved knowledge sentences. The comparative results for TED and WSeq-Sum on the Wizard of Wikipedia dataset are shown in Table 5.3 and Table 5.4, respectively.

The performance of both models (measured by Meteor and BLEU-2) initially increases with more injected knowledge sentences (e.g., top 1-3 sentences for TED), demonstrating that more relevant knowledge improves response generation. However, performance plateaus at a certain amount of knowledge (12 for the WSeq-Sum model, 3 for TED), likely due to the incorporation of more noise along with useful information. Diversity also follows a similar trend to relevance, with the plateau point varying (e.g., the WSeq-Sum model getting the highest BLEU-2 scores on 12 KS but highest diversity on 14 KS, as shown in Table 5.4).

The best-performing 3 KS TED and the 12 KS WSeq-Sum model are compared, finding that the former significantly outperforms the latter. Therefore, it can be concluded that the models can be enhanced with the right amount of external knowledge. Specifically, for the Wizard of Wikipedia dataset, the WSeq-Sum model performs best with 12 knowledge sentences injected, while for TED, the optimal number is 3.

5.4.3 The Impact of Noisy External Knowledge

Table 5.5: For the Original Reddit dataset: Compare different types of knowledge-grounded models on 3 typical metrics. TED is taken as the base model to do the significant test. * stands for significant test value $p < 0.05$.

Metrics	BLEU-2(%)	Meteor(%)	Div-2(%)
Seq2Seq-Attn	8.62*	6.70*	1.28*
Pointer-Generator	**12.80***	**8.76***	**15.20***
WSeq-Sum	8.52*	7.70*	2.00*
WSeq-Concat	8.67*	7.90*	2.80*
Multi-Task	8.98*	7.10*	5.29*
Transformer	8.62*	6.70*	1.30*
TED	11.89	8.40	3.80

In Section 5.3.1, It has been demonstrated that the Wizard of Wikipedia dataset has a significantly higher *UWR* than the Original Reddit dataset. Additionally, as shown in the previous section, the amount of injected knowledge can significantly affect the model performance, with noisy knowledge potentially degrading performance.

Table 5.5 displays the performance of all models on the noisy Original Reddit dataset[2]. It can be observed that when the injected knowledge sentences are noisier, the proposed TED model does not always perform the best. Particularly, the diversity score of the TED model is worse than that of the Pointer-Generator and Multi-Task models, indicating that the Pointer-Generator is more robust to noise compared to the proposed TED models, especially in terms of diversity. The TED model performs best only when a small amount of noise is infused.

5.5 Summary

A series of experiments were conducted on the Reddit and Wizard of Wikipedia datasets to assess the efficacy of various approaches for integrating external knowledge into generative models. State-of-the-art knowledge-injecting models were categorized into three classes: RNN-based, MemNN-based, and Transformer-based models. The results indicate that knowledge-injecting models generally outperform models that do not incorporate knowledge. Moreover, when comparing RNN-based and MemNN-based models, hierarchical RNN models exhibit significantly better performance in terms of both relevance and diversity. These findings highlight the sensitivity of knowledge-infusing models to noise, suggesting that higher-quality knowledge can lead to improved performance.

In addition to reviewing existing models, a novel approach named Transformer with Expanded Decoder (TED) was introduced for integrating additional information into the model. TED is built on the Transformer architecture and utilizes trainable parameters for context and knowledge represen-

[2] Note that the Original Reddit dataset adopts multi-reference as ground-truths, which leads to higher metric values. Therefore, the absolute values of different metrics are not directly comparable across different datasets. The multi-reference is not used for evaluation because the data distributions are different between the Original Reddit dataset and the Filtered Reddit dataset.

tations in the decoder, enabling weight tuning across various sources of evidence. The experiments demonstrate that TED is a highly effective knowledge-infusing model, especially when using a small amount of high-quality knowledge.

Given the sensitivity of most current models to noise and the observation that the TED model struggles in high-noise scenarios, the intention is to explore advanced retrieval methods for selecting highly relevant knowledge for generative models. Additionally, there is a plan to establish systematic and principled approaches for combining retrieval and generative models in the future work.

Chapter 6

Knowledge Selection for Knowledge-grounded Dialogue Generation

Chapter 5 confirms that incorporating knowledge can enhance dialogue generation, which is consistent with prior research (Weston et al., 2018; Ghazvininejad et al., 2018). However, noisy knowledge can introduce irrelevant information and degrade generative models (Zheng and Zhou, 2019). As a result, the response generation process requires an information retrieval component that must be optimized for selecting and injecting relevant knowledge into generative models. Evaluation of such approaches has shown that the knowledge based on posts alone may lack focus, i.e., may exhibit topic drifts and thus introduce noise. Table 6.1 illustrates Post-Retrieved Knowledge (i.e., retrieve knowledge sentences by taking posts as queries and be shortened as PRK) that has a good overlap with the post but introduces content that is not present in the response and is thus deemed non-relevant. By contrast, the Response-Retrieved Knowledge (RRK, is similar to PRK, but uses responses as queries) shares content with the response, thus illustrating that dialogue training needs to incorporate relevant knowledge related to the response.

The main difficulty in generating dialogues is selecting relevant response-related knowledge when the responses are unavailable during testing. To address this issue, a **T**ransformer & **P**ost based **P**osterior **A**pproximation (**TPPA**) method is proposed, which utilizes multiple processing stages of posts, post-related knowledge, and response-related knowledge to capture word and sentence-level characteristics (via word embeddings, Transformer, and max-pooling). These techniques can aid in ranking and selecting knowledge for new posts during the test phase. The overlap between true responses and TPPA outputs is shown in Table 6.1, using post-response pairs from the Wizard of Wikipedia dataset Dinan et al. (2019). Additionally, a piece of empirical evidence of TPPA's effectiveness is provided by incorporating TPPA-selected knowledge into generative models, specifically the Transformer Extended Decoder (TED), which facilitates knowledge integration from multiple sources. The TED and TPPA combination surpasses several strong baseline systems, including Post-KS (Lian et al., 2019) and SKT (Sequential Latent-knowledge Selection) (Kim et al., 2020), which do not separate knowledge selection from response generation modeling.

Table 6.1: Example of a post and a response from the Wizard of Wikipedia (WoW) data set with top 2 ranked outputs from TPPA, the post-retrieved knowledge PRK, and the response-retrieved knowledge RRK. Blue indicate words present in the WoW response and RRK but not in PRK.

WoW Post:	Yep. you've got to select for safety standards, of course, but when you're designing at a Mercedes level the folks buying those cars are going to expect a certain standard of comfort, too!
WoW Response:	Especially, I think consumers expect great in Formula One, highest class auto racing.
TPPA (top 1):	Formula One (also Formula 1 or F1 and officially the FIA Formula One World Championship) is the highest class of single seat auto racing that is sanctioned by the Federation Internationale de l'Automobile (FIA).
TPPA (top 2):	Stock car racing is a form of automobile racing found mainly and most prominently in the United States and Canada, with Australia, New Zealand and Brazil also having forms of stock car auto racing.
PRK (top 1):	Mercedes is part of the McQueen family and is the longest serving McQueen on the series.
PRK (top 2):	He also won races in midget cars, and sprint cars.
RRK (top 1):	Formula One (also Formula 1 or F1 and officially the FIA Formula One World Championship) is the highest class of single seat auto racing that is sanctioned by the Federation Internationale de l'Automobile (FIA).
RRK (top 2):	The FIA Formula One World Championship has been one of the premier forms of racing around the world since its inaugural season in 1950.

6.1 Motivations

Dialogue generation models that incorporate knowledge aim to expand the input beyond the observable post and incorporate a responder's knowledge. It is assumed that the available knowledge K_p for a given post P includes content that is related to the response, although the quality of that knowledge is not certain. The key issue is, thus, to determine which of the knowledge sentences $k \in K_p$ are relevant to the unobserved response R. During the training phase, where the post R, response R, and the corresponding knowledge set K_p are all available, P and R are used as queries to rank all the knowledge sentences in K_p and create the corresponding ranked lists: Response-retrieved Knowledge RRK and Post-retrieved Knowledge PRK, respectively. The lower-case rrk_1 and prk_1 are used to indicate the top 1 ranked item in RRK and PRK, respectively.

6.1.1 RRK Assessment on WoW Training Data

In this section, an analysis is conducted on RRK for the training data of Wizard of Wikipedia (WoW), where both the posts (P) and responses (R) are known, along with the corresponding knowledge set (K_p). It is assumed that a reasonable search algorithm is employed, and it is expected that rrk_1 (the top-ranked RRK) will exhibit a high degree of overlap with the response R that is used as a query. Additionally, it is also assumed that generative models will be capable of using rrk_1 to generate a response of good quality, given its overlap with the true response. The primary objective of this section is to gain insights into the potential difference that RRK can make when compared to the use of PRK alone.

Word count. A comparison is made between the number of common words (after removing stop words) between the original response (R) and four sequences: (1) the post P, (2) prk_1, i.e., the top 1 ranked item in PRK, (3) rrk_1, i.e., the top 1 ranked item in RRK, and (4) a random post chosen from the data set. Figure 6.1 displays the distributions of word overlaps. The count of

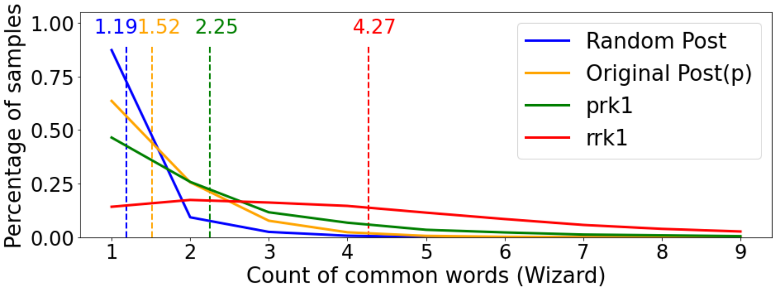


Figure 6.1: *Common word count distribution between each source and the target response on the WoW training set. The dashed lines are the average count of common words in each group (after removing stop words).*

common words is represented on the x-axis, while the percentage of the given sequences and responses R sample with the corresponding word overlap count is illustrated on the y-axis.

As expected, the word overlaps of P and prk_1 with R are similar, with the overlap of P and R being lower. For the randomly selected post P, the average term overlap with R is slightly lower but close to post P, suggesting that posts alone are not very informative for the response generation. The difference between prk_1 and rrk_1 is quite marked showing that rrk_1 has on average almost twice the overlap of the prk_1 (98% increase). Based on the Kolmogorov–Smirnov test [1], all the differences among the four groups in Figure 6.1 are statistically significant. For the Holl-E dataset, a similar trend can be observed.

Response generation. I assess the effectiveness of RRK when it is injected into the generative model by conducting experiments with the standard Transformer (Vaswani et al., 2017) and the Transformer with Expanded Decoder (TED) (Zheng and Zhou, 2019). Transformer takes only a post while TED uses a post and multiple sources of knowledge to get responses.

Table 6.2 (a) shows the results for Transformer with (1) original post, (2) a randomly selected sentence, (3) prk_1, (4) rrk_1 and (5) a human selected knowledge, i.e., a sentence provided in WoW. Table 6.2 with results metrics (BLEU-4, METEOR, and Div-2) show that replacing the original post with a

[1] https://en.wikipedia.org/wiki/Kolmogorov-Smirnov_test

randomly selected sentence reduces the performance significantly. Using prk_1 leads to lower performance, indicating a possible topic drift and noise. Using rrk_1 shows promising performance improvement; with higher retrieval performance, it may achieve the effectiveness of the human-selected knowledge. Similarly, for the TED generative model, the post content is incorporated, and the cumulative effect of adding knowledge from different sources is evaluated. As expected, the best performance is achieved by the human selection of knowledge followed by the RRK (Table 6.2 (b)).

Table 6.2: Injection of various sources into the Transformer and TED using WoW data set. All the values are percentages reported by the performance metrics (%).

(a) Transformer	BLEU-4	METEOR	Div-2
Original Post	1.76	6.6	7.3
Random Post	0.39	4.47	0.19
prk_1	1.23	6.36	5.62
rrk_1	**2.85**	**7.99**	**12.88**
Human selection	4.6	9.97	18.86
(b) TED	BLEU-4	METEOR	Div-2
Post+1 Random sentence	2.8	7.13	18.73
Post+prk_1	3.35	8.45	16.2
Post+rrk_1	**8.14**	**11.36**	**24.63**
Post+Human selection	10.06	13.13	25.7

In conclusion, it is worthwhile to put effort into creating resources that represent a responder's knowledge and effective retrieval methods to retrieve knowledge relevant to the response content. Since the responses are not available in the test phase, TPPA is devised to leverage post P and post-retrieved knowledge PRK and train models to approximate RRK.

6.2 TPPA Method

In this section, I will describe the architecture and the process of selecting knowledge using the TPPA method. Figure 6.2 depicts three TPPA components:

(1) **Post Processing Unit** comprising a word embedding and a Transformer that incorporates the post P and a set of n retrieved prk_i, where n is determined empirically (typically $n = 10$ out of 50 knowledge items in K_p, on av-

erage). The results are a Transformer representation v_p for the post and v_{PRK} for all of the *PRKs*. In the end, a single v_{prk} (representing the potentially most useful prk for identifying the rrk_1) is selected based on Auto-Pointer and Gumble Softmax algorithms.

(2) **Response Processing Unit** that, during training, considers each response R and corresponding K_p to get rrk_1 and a set of $negs$ (i.e., m negative samples which are non-relevant knowledge to the rrk_1) in order to train a word embedding that forms knowledge representation (called v_k). The number of negative examples m is selected empirically, to avoid over-fitting.

(3) **Knowledge Selection Unit**, a search component that uses v_p and v_{prk} as queries to score the knowledge representation v_k. The score is a weighted sum of similarity metrics using a hyper-parameter α that can be chosen to emphasize the similarity with P or prk.

TPPA operation consists of **Phase 1**: Training phase that utilizes training data (P, R, K_p) to train all the three components of the system based on known responses R; and **Phase 2**: Test phase during which individual post-knowledge samples (P, K_p) are processed in order to arrive at a selection of knowledge $(k \in K_p)$ to be injected into the generative models.

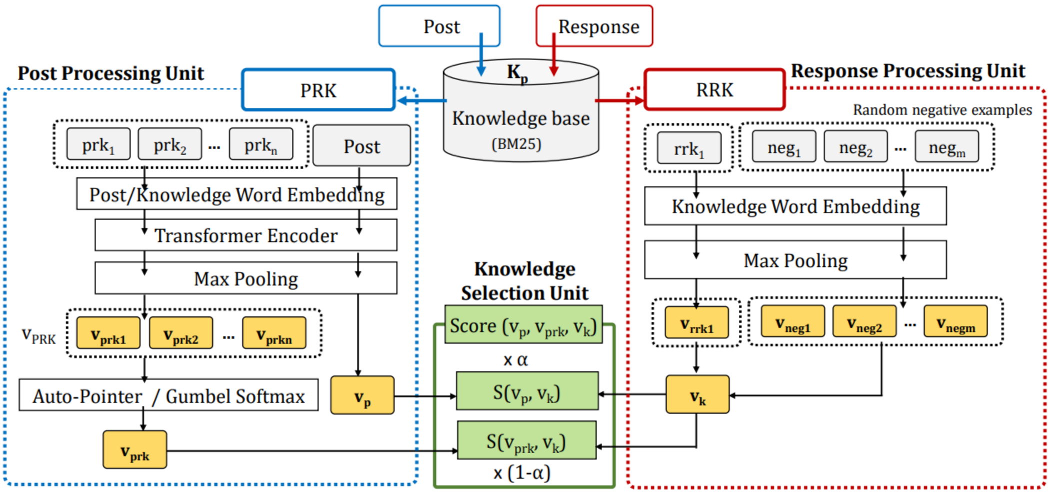


Figure 6.2: *TPPA Architecture comprises (1) Post Processing Unit, (2) Response Processing Unit (right), and (3) Knowledge Selection Unit (middle).*

6.2.1 TPPA Training phase

Post and PRK Processing

The post P and a set of $prk_i, i = 1, ..., n$ (i is the i-th ranked post-related knowledge) are processed with the same Transformer encoder to obtain word representations and then passed through the max-pooling to obtain the sequence semantic vector.

$$e(P) = \mathit{Transformer}_{\Theta}\left(e(w_i)\right), 1 \leq i \leq L \tag{6.1}$$

$$v_p = \mathit{maxpool}(e(P)) \tag{6.2}$$

where Θ is the trainable parameter set inside the Transformer. P is the input post, w_i is the *i-th* word of the P post sequence. L is the maximum post length. $e(w_i) \in \mathbb{R}^d$ is the post word embedding for w_i, and d is the embedding dimension. $e(P)$ represents the semantic representation of all the words in the post while v_p is the post representation (sentence-level). For the prk_i, they follow exactly the same process following Equation 6.1 and 6.2.

Multiple knowledge items prk_i are considered in order to construct an effective query for knowledge selection that complements the post and increases the chances of selecting knowledge that is relevant to the response. An auto-pointer is trained to assign scores to each prk_i. The auto-pointer module takes v_{PRK} as input and outputs a PRK scores vector (v_{ap}) that indicates the importance degree of the $prks$. This is followed by a Gumbel-Softmax (Jang et al., 2017) module to select the best $\boldsymbol{prk}$ for knowledge retrieval:

$$v_{ap} = (v_{PRK}W^T + b)W^T_{auto\_pointer} \tag{6.3}$$

$$v_{prk} = \mathit{Gumbel\text{-}Softmax}(v_{ap}, v_{PRK}) \tag{6.4}$$

where $v_{PRK} \in \mathbb{R}^{n \times d}$ represents all prk_i representations obtained by Eq. 6.1 and 6.2 and v_{prk} is the representation of the finally chosen post-related knowledge.

$W \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^{d}$ are trainable parameters; $W_{auto\_pointer} \in \mathbb{R}^{1\times d}$ is the trainable auto-pointer for selecting useful *prk*.

Response Processing Unit

The knowledge representation v_k is obtained by going through raw knowledge word embedding[2] and a max-pooling operation (seeing Figure 6.2 Response Processing Unit). The conduction of obtaining v_k is similar to Eq. 6.1 and 6.2 but replacing the Transformer to a raw knowledge word embedding lookup operation.

Since the objective is to augment vocabulary and avoid noise, during training, the positive knowledge is constrained to the highly relevant knowledge item, i.e., rrk_1 by using BM25. Knowledge is also randomly selected to serve as negative samples (from the union of all K_p after the rrk_1s of the posts are removed). Both the positive sample and negative samples will pass through the Response Processing Unit to gain their representations.

Knowledge Scoring and Selection

Following the post v_p and v_{prk} representation and knowledge representation v_k (here, v_k stands for an individual knowledge sentence), similarities $S(P, k)$ and $S(prk, k)$ are computed by:

$$S(P,k) = \frac{cosine(v_p, v_k)}{\|v_p\| \cdot \|v_k\|}; S(prk,k) = \frac{cosine(v_{prk}, v_k)}{\|v_{prk}\| \cdot \|v_k\|} \tag{6.5}$$

where $S(\cdot)$ designates the similarity function; v_p, v_k, and v_{prk} refer to the representations of the post, each knowledge unit, and the selected prk, respectively.

Depending on the type of dialogue, the response may incorporate the content of the post to a different degree. Thus, to support flexible scoring with regards to P and prk, a hyper-parameter α, is introduced to the final

[2] Alternative approaches, e.g., using Transformer based representations, were considered but led to sub-optimal results within the current TPPA setup.

scoring function:

$$Score(P, prk, k) = \alpha \times S(P, k) + (1 - \alpha) \times S(prk, k) \tag{6.6}$$

The α parameter is tuned on the training set and in the final $Score(P, prk, k)$, setting it to 0.7 to give more importance to the post.

After obtaining the scores of the positive and negative samples, for all the positive-negative sample pairs, the softmax function is calculated to obtain the similarity scores:

$$P(k_i|P, prk) = \frac{exp(\lambda Score(P, prk, k_i))}{\sum exp(\lambda Score(P, prk, k_i)} \tag{6.7}$$

calculating the probability of each k_i given the post P and the post-related-knowledge ($PRKs$). $k_i \in \{rrk_1; neg_1, neg_2, \ldots, neg_m\}$ are shown in the response processing unit in Figure 6.2, where $neg_1, \ldots, neg_m$ are m negative samples. λ is a smoothing factor of the softmax function and is a trainable parameter (Huang et al., 2013). The difference between the positive sample and the negative sample scores will be maximized by the equation.

$$Loss = \sum \left(-log(P(rrk_1|P) + \sum_j log(neg_j|P)) \right) \tag{6.8}$$

where $P(rrk_1|P)$ is the positive score, $P(neg_j|P)$ stands for the j-th negative score, where $1 \leq j \leq m$. m is the number of negative samples. During training, all of the trainable parameters, including the post-word embedding, Transformer architecture, auto-pointer, and knowledge word embedding, are updated by mini-batch gradient descent (the setup is in §5.2).

6.2.2 TPPA Test Phase

During the test phase, each new post P and corresponding K_p is processed using the Post Processing Unit and Response Processing Units, with parameters obtained during the training phase. Each knowledge k_i and its corresponding post are scored using the $Score(P, prk, k_i)$ (Eq. 6.6) and TPPA returns the final rank of the knowledge candidates.

6.3 Experiment

The proposed approach for knowledge injection separates the *knowledge selection* from the *response generation* models. We, thus, evaluate TPPA in terms of (1) precision in selecting relevant knowledge for a given post, judged by whether the rrk_1 can be ranked within the top n position, and (2) effectiveness of the retrieved knowledge when injected into a response generation model.

6.3.1 Datasets

Following the previous work (Kim et al., 2020) setting in which the knowledge selection was conducted, the experiments are also done on two publicly available data sets: Wizard of Wikipedia (WoW, Dinan et al. (2019)) and Holl-E (Moghe et al., 2018) datasets. Both of them contain human annotations by employing Amazon Mturk workers to interact with each other. Details are introduced in Sec. 2.3.

6.3.2 Implementation Details

In the experiments, the dimension of word embedding is 300, and the multi-head number of the Transformer is 4. The vocabulary is obtained by ranking the training data by word frequency, with the size of 50,000 top frequent terms selected. The minimum post length is set to 8 tokens (Zheng and Zhou, 2019). Each knowledge item is represented by a sentence. During model training, a mini-batch size of 64 is used. Adam optimizer is used for optimization (Kingma and Ba, 2015). The initial learning rate is set to 0.001 and halved when reaching the plateau (decreasing patience is set to 2 epochs). All the experiments are run on a single TITAN V GPU. The TPPA model requires 2 hours to train on the WoW data set.

6.3.3 Metrics

The quality of the generated responses is evaluated using five standard metrics: BLEU (Papineni et al., 2002), Meteor (Banerjee and Lavie, 2005), and Bert-Score (BS) (Zhang et al., 2019b) that are based on co-occurrence of n-grams between the system response and the ground-truth, calculating the token similarity using contextual embeddings. In this work, the BS version being used is *roberta-large_L17_idf_version=0.3.3(hug_trans=2.8.0)*[3]; Diversity scores (Div-2) (Li et al., 2016a) calculates the proportion of distinct bi-grams out of all the distinct words.

For knowledge selection, I use $P@n$ that calculates the precision at a given rank n, measuring whether the ground truth (rrk_1) exists within the top n retrieved knowledge.

6.3.4 Baselines

For comparison, the TPPA knowledge selection on the retrieval performance will be compared to three baseline models:

BM25 (Robertson and Walker, 1994) is an unsupervised probabilistic retrieval algorithm, which is robust for short document (sentence) retrieval.

DrQA (Chen et al., 2017) uses bigram hashing and TF-IDF matching with a multi-layer recurrent neural network model.

CNN-DSSM (Shen et al., 2014) uses CNN for semantic matching of queries and documents.

In order to evaluate the effectiveness of the selected knowledge for *response generation*, TPPA output is compared with three models:

[3] https://github.com/Tiiiger/bert_score

WSeq (Tian et al., 2017) uses weighted sum and concatenation of the post and its contextual utterances, and obtain representations through an RNN.
MemNet (Ghazvininejad et al., 2018) leverages a multi-task learning framework to jointly train 'post-to-response', 'knowledge-to-response' and 'knowledge-to-knowledge' tasks for response generation.
TED (Zheng and Zhou, 2019) adopts Transformer as the backbone framework to inject knowledge by assigning weights to the knowledge from multiple sources.

Finally, I consider two methods that jointly train the knowledge selection model and dialogue generation model, and use them in both sets of experiments:
Post-KS (Lian et al., 2019) approximates posterior-distribution of knowledge, i.e., $p(k|P, R)$ using prior-distribution $p(k|P)$ and jointly train a knowledge selection model and a dialogue generation model.
SKT (Kim et al., 2020) takes into account context from multi-turn dialogues (current action and 2 prior turns) and considers knowledge selection as a sequential decision process.

6.4 Experimental Results and Analysis

6.4.1 Knowledge Selection Evaluation

For the TPPA method, the quality of the selected knowledge is determined by the embedding parameters obtained during the training phase. They are, in turn, related to the knowledge resources used for training (Response Processing Unit) and the quality of the transformer representation of P and prk (Post Processing Unit), shown in Figure 6.2. The resources are constructed from individual knowledge sets K_p. For each training sample, it consists of a post P, a rrk_1 (i.e. the top 1 ranked response-retrieved knowledge), n $prks$ (i.e. the top n ranked post-retrieved knowledge), and m $negs$ (i.e. randomly

Table 6.3: Retrieval precision on the WoW and Holl-E data sets. ‘*’ means t-test $p < 0.05$ compared with the baseline $BM25$; ‘†’ is the $p < 0.05$ compared with the best performing group. **Bold** indicates the best performance group when changing the number of negative samples. Underline indicates the best group among all methods.

Exp Model		Wizard of Wikipedia (%)		
		P@1	P@5	P@10
BM25		4.9†	18.6†	31.1†
DrQA		4.1†	13.6*†	21.7*†
CNN-DSSM		8.2*†	31.3*†	48.8*†
Post-KS		6.2*†	-	-
SKT		9.01*	-	-
TPPA	1rrk-1neg-10prk	8.9*†	33.0*†	49.2*†
	1rrk-4neg-10prk	10.0*	36.5*†	54.5*
	1rrk-10neg-10prk	9.8*	36.4*†	54.2*†
	1rrk-20neg-10prk	10.1*	37.8*	55.0*
	1rrk-30neg-10prk	**10.1***	**38.0***	**55.1***
	1rrk-40neg-10prk	8.2*†	31.3*†	48.2*†
TPPA	1rrk-30neg-1prk	10.2*	38.4*	55.1*
	1rrk-30neg-10prk	10.1*	38.0*	55.1*
	1rrk-30neg-20prk	10.0*	37.3*†	55.1*
	1rrk-30neg-30prk	9.7*	35.2*†	52.4*†

Exp Model		Holl-E (%)		
		P@1	P@5	P@10
BM25		10.5†	33.4†	48.5†
DrQA		13.3*†	29.4*†	35.4*†
CNN-DSSM		15.2*†	34.9*†	50.0†
Post-KS		5.5*†	-	-
SKT		11.6*†	-	-
TPPA	1rrk-1neg-10prk	13.6*†	37.0*†	51.3*†
	1rrk-4neg-10prk	15.5*†	38.3*†	52.7*†
	1rrk-10neg-10prk	**16.6***	**40.4***	**54.5***
	1rrk-20neg-10prk	14.8*†	36.9*†	51.1†
	1rrk-30neg-10prk	15.7*†	39.1*†	53.2†
	1rrk-40neg-10prk	16.2*	39.5*	53.2
TPPA	1rrk-10neg-1prk	16.3*	39.0*†	52.7*†
	1rrk-10neg-10prk	16.6*	40.4*	54.5*
	1rrk-10neg-20prk	16.6*	39.0*	52.9*†
	1rrk-10neg-30prk	15.4*†	38.6*	52.7*†

chosen m sentences). Thus, $1rrk$-$1neg$-$10prk$ indicates that the rrk_1, 1 random knowledge item, and top 10 $prks$ are selected for each P. In the test phase, monitoring is done to determine whether, for a new post P in the test set, different retrieval models rank its corresponding ground truth, i.e., rrk_1 for P, within the top 1, 5, or 10 ranked items.

Results in Table 6.3 show that: **(1)** TPPA provides at least one model that outperforms all other models on the WoW and Holl-E data sets, on all three metrics P@1, P@5, and P@10. **(2)** The composition of the knowledge base

Table 6.4: Performance of generative models MemNet, WSeq, and TED with the best TPPA knowledge selection. Post-KS and SKT rely on their jointly trained models. BS refers to Bert-Score.

Exp Model	Wizard of Wikipedia (%)			
	BLEU-4	METEOR	Div-2	BS
MemNet	1.24	6.39	2.24	81.5
WSeq	2.13	7.17	13.29	82.86
Post-KS	1.35	5.96	22.32	81.3
SKT	3.14	7.29	27.8	83.4
TED	3.91	8.82	18.16	82.9
Exp Model	Holl-E (%)			
	BLEU-4	METEOR	Div-2	BS
MemNet	5.59	7.63	0.18	84.6
WSeq	5.9	7.94	3.63	83.71
Post-KS	3.79	5.98	2.41	81.3
SKT	9.16	8.48	22.9	82.9
TED	12.66	10.37	17.95	84.1

affects the TPPA knowledge selection: for the WoW data set and fixed number of 10*prk*, increasing the number of *neg* items improves the performance until reaching its plateau at 1*rrk*-30*neg*-10*prk*; for the Holl-E data set, the best combination is 1*rrk*-10*neg*-10*prk*. **(3)** For a fixed number of *neg*, I vary the number of *prks* items and find that: (i) for WoW and n=30, the optimal *prk* number is 1; and (ii) for Holl-E and *neg*=10 the optimal *prk* number is 10.

Based on these findings, 1*rrk*-30*neg*-1*prk* is used for WoW and 1*rrk*-10*neg*-10*prk* for Holl-E as the knowledge selection models. The newly ranked knowledge set by TPPA will be used to experiment with MemNet, WSeq, and TED models on the response generation task.

6.4.2 Response Generation Evaluation

The initial set of experiments is conducted to assess the robustness of the generative models (Table 6.4) and find that: (i) SKT and TED models outperform others, (ii) MemNet have unstable performance and constantly under-performs on Div-2. Furthermore, since SKT and Post-KS cannot inject multiple knowledge items, for further discussion, WSeq and TED are chosen for experiments. They are combined with knowledge selection from (i) BM25, (ii) SKT (single knowledge item), (iii) CNN-DSSM (supervised search algorithm on post only),

Table 6.5: Knowledge-injection results on the Wizard of Wikipedia data set. The values are percentages (%). '*' means the t-test $p < 0.05$ compared with the BM25 algorithm. 'Top 1', 'Top 5', and 'Top 10' denotes injecting top 1 or 5, or 10 ranking knowledge. BS is Bert-Score. **Bold** indicates the best score apart from the rrk_i group.

TED+Top 1	BLEU-4	METEOR	Div-2	BS
BM25	3.35	8.45	16.2	82.7
SKT	**4.05***	**8.82***	18.8*	82.8*
CNN-DSSM	3.5	8.62	**20.08***	82.8
TPPA	3.91*	**8.82***	18.16	**82.9**
rrk_1	8.14*	11.36*	24.63*	84.3*
TED+Top 5	BLEU-4	METEOR	Div-2	BS
BM25	3.17	7.81	**18.33**	82.99
CNN-DSSM	3.81	8.82	16.98	83.16
TPPA	**3.88***	**8.97***	17.22*	**83.23**
rrk_5	4.99*	10.49*	19.04*	83.7*
TED + Top 10	BLEU-4	METEOR	Div-2	BS
BM25	3.01	7.98	15.7	83.2
CNN-DSSM	**3.59***	8.98*	**14.8***	83.38
TPPA	3.53*	**9.09***	14.66*	**83.4***
rrk_{10}	4.05*	9.56*	15.87*	83.6*
WSeq+Top 1	BLEU-4	METEOR	Div-2	BS
BM25	1.94	6.98	12.96	82.76
SKT	2.0	7.02	**13.73**	82.8
CNN-DSSM	2.04	7.07	13.25	82.81
TPPA	**2.13**	**7.17***	13.29	**82.86**
rrk_1	2.23*	7.35*	13.23	83.0*
WSeq+Top 5	BLEU-4	METEOR	Div-2	BS
BM25	2.05	7.18	17.59	82.85
CNN-DSSM	2.07	7.37	18.32	83.03*
TPPA	**2.15***	**7.57***	**18.55***	**83.1***
rrk_5	2.61*	8.0*	18.75*	83.3*
WSeq + Top 10	BLEU-4	METEOR	Div-2	BS
BM25	2.31	7.44	19.48	83.0
CNN-DSSM	2.44	7.88*	**20.19**	83.3*
TPPA	**2.59**	**7.97**	19.72	**83.35**
rrk_{10}	3.01*	8.67*	21.07	83.66*

(iv) TPPA using both post and post-retrieved knowledge items, and (v) rrk_i (i means top i ranked response-retrieved knowledge, it is set to be 1, 5 and 10 in this work), to determine the upper bound when responses are known). The comparisons for the two data sets are shown in Table 6.5 and Table 6.6.

We can observe that: **(1)** Injecting knowledge from SKT, CNN-DSSM, and TPPA generally outperforms the post-only selection using BM25 (Table 6.5 and 6.6) on both the WoW and Holl-E data sets in terms of the BLEU-4, METEOR, and Bert-Score. TED performance suffers from increased knowledge injection. Indeed, for TED + rrk_i, i.e., using 'perfect knowledge' the performance decreases with the increasing number of knowledge items. Zheng and Zhou (2019) claim that TED lacks a noise-filtering mechanism and thus underperforms with too much data. **(2)** Not surprisingly, knowledge selection methods with better retrieval performance achieve better response generation metrics. We can see from Table 6.5 and 6.6 and the corresponding retrieval performance in Table 6.3. For the WoW data set, the TPPA with $1rrk$-$30neg$-$1prk$ achieves the best retrieval performance and better results (Table 6.5) on both generative models (TED and WSeq) across different settings. This is confirmed on the Holl-E data set (Table 6.6) where TPPA outperforms other models, including Post-KS and SKT. This confirms the conjecture that improving retrieval for knowledge injection should improve response generation.

6.4.3 Upper-bound Analysis

The upper bound for knowledge selection is the rrk_i group. We can observe how all of the retrieval models perform in combination with TED and WSeq (Table 6.5 and 6.6). For the sake of concreteness, the BLEU-4 metric is utilized for illustration. Table 6.5 and 6.6 show that low levels of knowledge-injection, e.g., a single knowledge item (Top 1), leads to large differences between TPPA and RRK in BLEU-4: 4.23% (8.14%-3.91%) for WoW and 33.28% (45.94%-12.66%) for Holl-E data set. Despite that, TPPA manages to better approximate RRK than other models and improves response generation.

6.4.4 Analysis of Added Useful Words

In order to analyze the properties of the generated responses, a metric, called *Useful Word Overlapping Rate* (UWOR) is defined. As illustrated in Sec. 2.4, if a word appears in the response but not in the post, it is a useful word. UWOR measures the coincidence ratio of two sequences and can be formulated as

$$UWOR(S_1, S_2) = \frac{overlap(S_1, S_2)}{distinct(S_2)}$$

Table 6.6: Knowledge-injection results on the Holl-E data set. The values are percentages (%). '*' means the t-test $p < 0.05$ compared with the BM25 algorithm. 'Top 1', 'Top 5', and 'Top 10' denotes injecting top 1 or 5, or 10 ranking knowledge. BS is Bert-Score. **Bold** indicates the best score apart from the rrk_i group.

TED+Top 1	BLEU-4	METEOR	Div-2	BS
BM25	9.87	9.09	**26.21**	83.6
SKT	9.01	8.56	19.86*	83.4*
CNN-DSSM	11.56*	9.84*	23.51*	83.9
TPPA	**12.66***	**10.37***	17.95*	**84.1***
rrk_1	45.94*	30.61*	29.03*	89.6*
TED+Top 5	BLEU-4	METEOR	Div-2	BS
BM25	11.4	10.22	**24.16**	83.9
CNN-DSSM	12.02	10.4	23.71	84.0
TPPA	**12.92***	**11.12***	17.87*	**84.2**
rrk_5	21.81*	17.15*	24.96*	85.9*
TED + Top 10	BLEU-4	METEOR	Div-2	BS
BM25	5.5	**8.36**	2.45	83.5
CNN-DSSM	5.39	8.24	**2.6***	**83.6**
TPPA	**5.6**	8.24	2.53*	**83.6**
rrk_{10}	6.53*	9.88*	2.75*	84.0*
WSeq+Top 1	BLEU-4	METEOR	Div-2	BS
BM25	4.58	7.25	4.33	83.68
SKT	5.81*	7.77*	3.09	83.6*
CNN-DSSM	5.6*	7.62*	**4.48***	83.5*
TPPA	**5.9***	**7.94***	3.63*	**83.71**
rrk_1	6.5*	8.95*	4.6*	83.97*
WSeq+Top 5	BLEU-4	METEOR	Div-2	BS
BM25	5.15	7.51	8.65	83.43
CNN-DSSM	5.53*	7.69	**9.78***	83.17*
TPPA	**5.96***	**7.74***	7.82*	**83.59***
rrk_5	7.22*	9.55*	9.39*	83.85*
WSeq + Top 10	BLEU-4	METEOR	Div-2	BS
BM25	5.28	7.15	13.85	83.43
CNN-DSSM	5.88*	7.35*	**16.26***	83.3*
TPPA	**5.89***	**7.43***	12.43*	**83.7***
rrk_{10}	8.19*	10.41*	15.73*	84.3*

where, S_1 and S_2 represent two sequences. Here in this work, $UWOR(P, R)$ is represented to be a Useful Word Overlapping Rate between the post and response (P for the post and R for the response). The $overlap(\cdot)$ is the number of distinct overlapping useful words between two sequences. $distinct(\cdot)$ is a distinct number of words. The stop words of the two sequences are removed before calculating UWOR.

Whether the retrieved knowledge brings additional useful words is further examined. The calculation *UWOR(k-P, R)* is done for that purpose, where *k-P* is a set of words in the knowledge ($k \in K_p$) but not in the associated post P, i.e., $\{w|w \in k \cap w \notin P\}$, w is the word of a sequence.

The results are shown in Table 6.7. For each experiment group in Table 6.7, the top 1 ranked sentence is selected for calculation. *UWOR(P, R)* values for the WoW and Holl-E data sets are just 14.6% and 7.52%, respectively. Considering the TPPA, for WoW the number of additionally added useful words is comparable to what the post brings (10.25% vs. 14.6%); for the Holl-E, the retrieved knowledge brings more than double the useful words than the post (15.98% vs. 7.52%). This demonstrates the effectiveness of TPPA which can expand additional useful words from knowledge.

Table 6.7: The useful word overlapping rate results of WoW and Holl-E data sets. All values are shown as percentages (%).

Exp Name		Wizard of Wikipedia	Holl-E
UWOR(p, r)		14.6	7.52
UWOR$(k-p,r)$	BM25	4.11	9.42
	SKT	9.0	9.52
	CNN-DSSM	9.32	14.92
	TPPA	**10.25**	**15.98**
	rrk_1	34.52	67.84

6.5 Summary

By investigating the knowledge associated with post-response pairs, we can gain valuable insights into how the performance of generative models can be improved through the selection of response-retrieved knowledge (RRK). However, since the response is not observable during the test phase, a TPPA method is developed, which carefully embeds knowledge and optimizes the representation of the post and post-related knowledge (PRK) to select knowledge items. The empirical results demonstrate the superiority of TPPA, which can be used separately from the generative models, allowing for the exploration of alternative components and models.

While effective, a potential limitation of the TPPA model must be addressed: the quality of the knowledge base significantly impacts its effectiveness. The WoW and Holl-E datasets used for experimentation contained high-quality candidate knowledge items that were manually selected. The analysis of the WoW dataset shows that the rrk_1 group contains, on average, more than two common words compared to the prk_1 group, which could help in forming the ground truth response. A similar trend is observed in the Holl-E dataset.

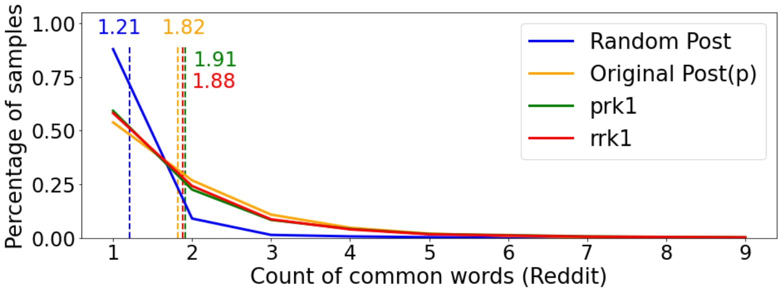


Figure 6.3: *Common word count distribution between each source and the target response on the Reddit training set. The dashed lines are the average count of common words in each group (after removing stop words).*

When examining the Reddit dataset[4], we can see in Figure 6.3 that the rrk_1 group and prk_1 group contain nearly the same number of common words as the ground-truth response. This is not unexpected, as Reddit is an online forum where each post typically includes a URL to a web page (grounding) that defines the post's topic, provided by the author. However, the responders may not read that information and instead respond based on their own knowledge. Empirically, it can be found that TPPA cannot benefit from such knowledge and performs worse than the baselines. This suggests that when the quality of knowledge is potentially low, using PRK as evidence for pseudo-relevance feedback may result in topic drift.

The TPPA model proposed in this study can perform knowledge selection by treating it as a retrieval task, where it ranks knowledge sentences based on assigned importance weights. This is done at the sentence level, meaning that the model views the knowledge sequence as a whole without distinguishing the importance of individual words within the sequence. However, the words in the same sequence should also be viewed with different importance. Given this consideration, my aim is to investigate whether word-level weighing would improve dialogue generation.

[4] https://github.com/mgalley/DSTC7-End-to-End-Conversation-Modeling

Chapter 7

Knowledge-Grounded Dialogue Generation with Term-level De-noising

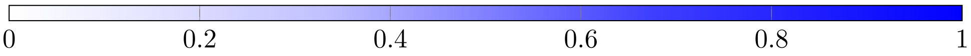
In the previous Chapters, two types of generative models for dialogue have been discussed: context-aware dialogue generation and knowledge-grounded dialogue generation. Context-aware dialogue generation models, as presented by Serban et al. (2016); Tian et al. (2017); Zhang et al. (2019a), aim to enhance response generation by utilizing contextual information from the conversation. On the other hand, knowledge-grounded dialogue generation models, as explored by Zheng and Zhou (2019); Kim et al. (2020); Dinan et al. (2019), focus on improving the performance of generative models by leveraging relevant external knowledge sources.

To my best knowledge, all previous methods have primarily concentrated on selecting and integrating knowledge at the sentence or paragraph level. However, this approach can lead to difficulties in managing potential noise, such as the inclusion of irrelevant words or phrases. Previous research studies by Galley et al. (2019), Zheng et al. (2020) have demonstrated that introducing noise can negatively impact the quality of response generation. Therefore, it is crucial to explore how to modify the impact of individual terms in the selected knowledge in order to reduce noise and improve the generated responses. This issue has not been systematically addressed in prior studies, which suggests that there is an opportunity to further refine knowledge-grounded dialogue generation models. By developing techniques that can better filter out the noise and focus on relevant terms within the chosen knowledge, researchers may be able to create more accurate, coherent, and contextually appropriate responses in dialogue generation tasks.

Table 7.1: Example of a post, ground truth response, injected knowledge, and generated response by the Knowledge Term Weighting Model (KTWM). The term highlights indicate the predicted probability of a term being useful.

Post: I am a big fan of education. I think people don't realize how important it is.
Ground-truth response: Sure, education is important since it facilitates learning and the acquisition of skills.

Knowledge terms weighted by KTWM:
Education is the process of facilitating learning , or the acquisition of knowledge , skills , values , beliefs , and habits
Response generated by KTWM: I agree. Education is a great way to learn about facilitating learning.



This work fills this gap by introducing a novel *Knowledge Term Weighting Model* (KTWM) for dialogue generation, which effectively estimates term weights of the injected knowledge and incorporates such weights into the response generation. The response generation thus benefits from such nuanced term-level knowledge weighting, promoting important knowledge terms rather than treating equally all the terms in the selected sentences. In Table 7.1, an example of the KTWM term weighting and its generated response are given: the terms 'education', 'is', 'facilitating', and 'learning' are given higher weights correctly as they do appear in the ground-truth response, while the words 'values' and 'beliefs' are correctly assigned lower scores.

An extensive range of experiments with KTWM are conducted on two publicly available datasets: Wizard of Wikipedia (with seen and unseen test topics) (Dinan et al., 2019) and Holl-E (Moghe et al., 2018). KTWM performs consistently well with different selections of knowledge, specifically with Post-

KS (Lian et al., 2019), SKT (Sequential Latent-Knowledge Selection) (Kim et al., 2020) and TED (Transformer with Expanded Decoder) (Zheng and Zhou, 2019). The approach in this work achieves both a superior performance in knowledge-grounded dialogue generation and new insights into the impact of the knowledge term weighting on that performance.

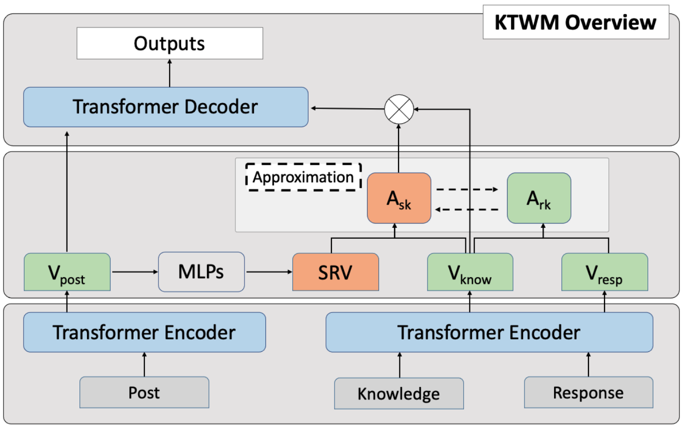


Figure 7.1: *Overview of the Knowledge Term Weighting Model. V stands for representations for post, knowledge, and response. SRV means simulated response vector, and A_{sk} and A_{rk} denote knowledge terms' weights matrices assigned with SRV and response, respectively.*

7.1 Methodology

In this section, the basic concepts will be introduced, and the proposed method KTWM for the term-weighting of the injected knowledge will be described in detail. Assuming a collection of posts P and responses R, there exists a collection K_p of knowledge sets with sentences relevant to the specific post-response pair (P, R). For a given pair (P, R), the knowledge injection process consists of three stages: (1) knowledge selection, (2) knowledge term-weighting, and (3) decoding with the weighted knowledge terms. The primary focus here is on stage (2), which concerns the effectiveness of term-weighting for the knowledge incorporated in the KTWM.

To start, an overview of the KTWM is presented as depicted in Figure 7.1. During the training phase, the post, knowledge, and response are transformed into vectors, denoted as V_{post}, V_{know}, and V_{resp}, respectively. Subsequently, V_{post}

undergoes processing through Multi-Layer Perceptrons (MLPs) to produce the Simulated Response Vector (SRV), which serves as a pseudo response during the test phase since the actual ground truth responses are not available. The matrices A_{sk} and A_{rk} represent the weight matrices for knowledge terms, and they are determined by the SRV and the response, respectively. The key of the KTWM lies in the approximation module, where these two matrices associated with knowledge terms are approximated. This approximation aims to bring the SRV closer to the ground truth response. In the subsequent sections, we will delve into the detailed architecture presented in Figure 7.2, as well as explore the KTWM decoder, illustrated in Figure 7.3.

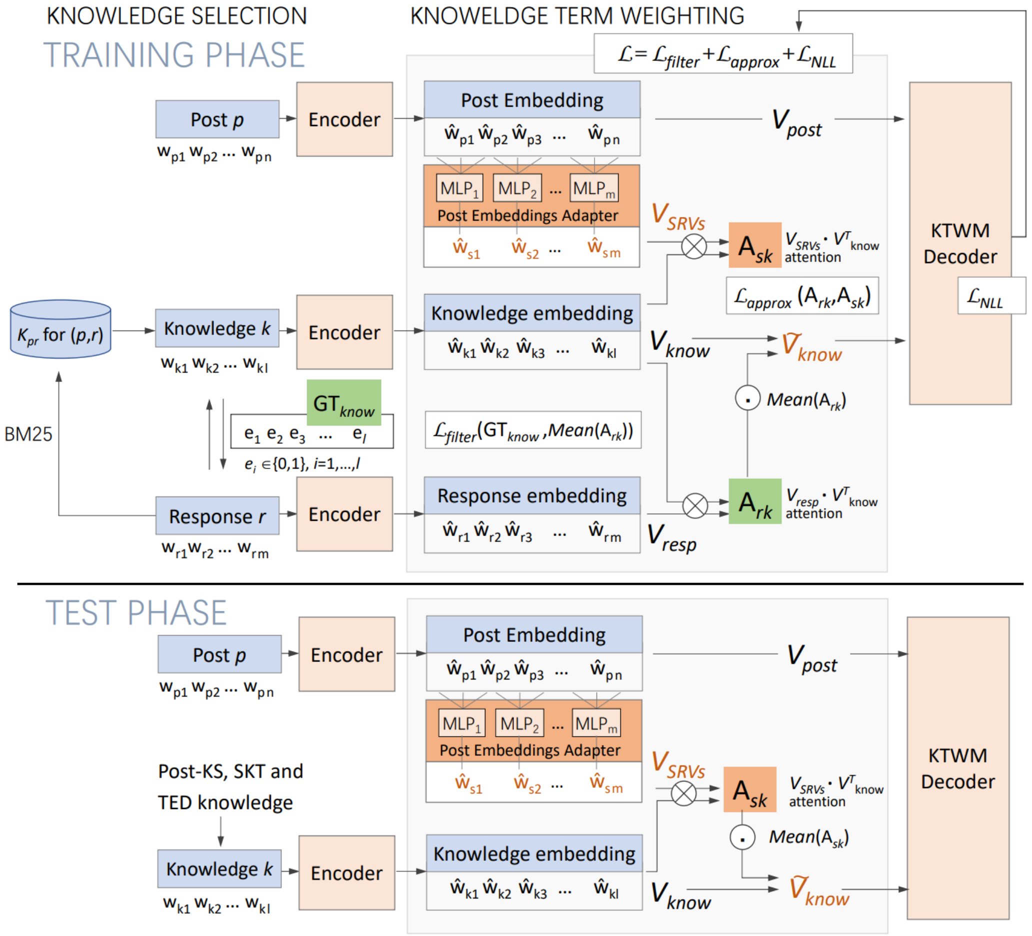

KNOWLEDGE SELECTION
KNOWELDGE TERM WEIGHTING
TRAINING PHASE
$\mathcal{L} = \mathcal{L}_{filter} + \mathcal{L}_{approx} + \mathcal{L}_{NLL}$
Post p
$w_{p1}\ w_{p2} \cdots w_{pn}$
Encoder
Post Embedding
$\hat{w}_{p1}\ \hat{w}_{p2}\ \hat{w}_{p3} \cdots \hat{w}_{pn}$
V_{post}
MLP_1 MLP_2 … MLP_m
Post Embeddings Adapter
$\hat{w}_{s1}\ \hat{w}_{s2} \cdots \hat{w}_{sm}$
V_{SRVs}
A_{sk}
$V_{SRVs} \cdot V^T_{know}$ attention
$\mathcal{L}_{approx}(A_{rk}, A_{sk})$
KTWM Decoder
$\mathcal{L}_{NLL}$
K_{pr} for (p,r)
Knowledge k
$w_{k1}\ w_{k2} \cdots w_{kl}$
Encoder
Knowledge embedding
$\hat{w}_{k1}\ \hat{w}_{k2}\ \hat{w}_{k3} \cdots \hat{w}_{kl}$
V_{know}
$\tilde{V}_{know}$
$Mean(A_{rk})$
GT_{know}
BM25
$e_1\ e_2\ e_3 \cdots e_l$
$e_i \in \{0,1\}, i=1,\ldots,l$
$\mathcal{L}_{filter}(GT_{know}, Mean(A_{rk}))$
Response r
$w_{r1} w_{r2} \cdots w_{rm}$
Encoder
Response embedding
$\hat{w}_{r1}\ \hat{w}_{r2}\ \hat{w}_{r3} \cdots \hat{w}_{rm}$
V_{resp}
A_{rk}
$V_{resp} \cdot V^T_{know}$ attention
TEST PHASE
Post p
$w_{p1}\ w_{p2} \cdots w_{pn}$
Encoder
Post Embedding
$\hat{w}_{p1}\ \hat{w}_{p2}\ \hat{w}_{p3} \cdots \hat{w}_{pn}$
V_{post}
MLP_1 MLP_2 … MLP_m
Post Embeddings Adapter
$\hat{w}_{s1}\ \hat{w}_{s2} \cdots \hat{w}_{sm}$
V_{SRVs}
A_{sk}
$V_{SRVs} \cdot V^T_{know}$ attention
$Mean(A_{sk})$
KTWM Decoder
Post-KS, SKT and TED knowledge
Knowledge k
$w_{k1}\ w_{k2} \cdots w_{kl}$
Encoder
Knowledge embedding
$\hat{w}_{k1}\ \hat{w}_{k2}\ \hat{w}_{k3} \cdots \hat{w}_{kl}$
V_{know}
$\tilde{V}_{know}$

Figure 7.2: *Architecture of the Knowledge Term Weighting Model (KTWM) showing the operations in the training and test phase. ⊗ designates matrix multiplication; ⊙ designates element-wise multiplication.*

7.1.1 Knowledge Selection and Representation

I represent each post P, response R, and an individual knowledge sentence $k \in K_p$ as a vector of terms. The set K_p typically contains multiple knowledge sentences. The BM25 retrieval method is used to rank the sentences by their relevance to the post (in the test phase) or response (in the training phase). For knowledge injection, the top-ranked sentences will be considered. When the knowledge injection requires a specific number of terms to be used, additional sentences from the ranked list will be used to meet that requirement.

When a knowledge sentence k is retrieved based on a response R as a query, a ground truth vector GT_{know} is defined for the knowledge k with the weight of 1 assigned to the knowledge terms that are present in R and the weight of 0 assigned to those that are not, i.e., $GT_{know} = (e_1, e_2, \ldots, e_l)$, where $e_i \in \{0, 1\}$, $i = 1, \ldots, l$.

Encoders. Transformer (Vaswani et al., 2017) is adopted as the backbone framework for the training and testing of KTWM. The Transformer encoder consists of a self-attention layer and a transition layer involving the layer normalization and residual network. More details about the Transformer are illustrated in Sec. 2.1.2.

Figure 7.2 shows transformer encoders (*encoders* for short) used for the post, knowledge, and response representations and processing. w is used to designate an original term and $\widehat{w}$ to designate the term's representation. In Figure 7.2, n, m, and l are three pre-defined hyper-parameters that refer to the length of the post (P), response (R), and knowledge (k), respectively (e.g. w_{pi} means the i-th term of the post). Any sequence that is longer or shorter than the given length will be truncated or padded to the given length. By applying the encoder

$$V_{post} = Encoder(w_i)(i \in [1, n]) \tag{7.1}$$

the post terms representations is denoted as V_{post}, comprising $\widehat{w}_{p1}, \widehat{w}_{p2}, \ldots, \widehat{w}_{pn}$

(in Figure 7.2), from the original terms $w_{p1}, w_{p2}, \ldots, w_{pn}$. Similarly to V_{post} in Eq. (7.1), V_{know} and V_{resp} can be gained as term representations of the corresponding knowledge and the response, respectively.

7.1.2 Knowledge Term Weighting

The fundamental premise of this work is that knowledge terms related to or present in the response should be more effective in improving dialogue generation. Thus, it can be beneficial to use methods such as attention distribution of response and knowledge embeddings to determine the weights of individual knowledge terms. However, in the real setting and during the test phase, we can only use terms and knowledge related to the post. Furthermore, the post embeddings can significantly differ from the response ones. Thus, assigning weights to the knowledge terms based on their similarity to post embeddings is unlikely to be sufficient (Xing et al., 2018).

For that reason, the goal is to learn how to transform the post embeddings to be effective in knowledge-term weighting. This is achieved by training a *Post Embeddings Adapter* that can, for a new post, generate *Simulated Response Vectors* (SRVs) and use them in place of the response vectors to score post-related knowledge terms.

To that effect, a set of Multi-Layer Perceptrons (MLPs) is designed:

$$MLP = \sum_{i=1}^{n} \widehat{w}_{pi} W_i + b \tag{7.2}$$

$$\widehat{w}_{sj} = MLP_j(\widehat{w}_{p1}, \widehat{w}_{p2}, \ldots, \widehat{w}_{pn})(j \in [1, m]) \tag{7.3}$$

where W_i and b are trainable parameters for each term p_i of the post p; $\widehat{w}_{sj}$ is the representation of the j-th term of the simulated response vector (SRV). The number of MLPs is the same as the number of terms in a given response.

During the training phase, MLPs learn the transformation of the post embeddings into SRVs that capture the ground truth response representation for a given post p. SRVs are then used to assign appropriate weights to the knowledge terms when response information is not available.

SRVs Approximation and Training. The training phase begins with V_{post}, V_{know}, V_{resp} and randomly initiated parameters of MLPs to produce the initial set of V_{SRVs} for a given post. Each iteration then involves comparisons of (a) the response embeddings V_{resp} and knowledge embeddings V_{know}, and (b) SRVs with the knowledge embeddings V_{know}. More precisely, the term-wise attention distributions A_{rk} and A_{sk} are computed:

$$A_{rk} = sigmoid(V_{resp}V_{know}^{T}) \tag{7.4}$$

$$A_{sk} = sigmoid(V_{SRVs}V_{know}^{T}) \tag{7.5}$$

where $A_{rk} \in \mathbb{R}^{m \times l}$ and $A_{sk} \in \mathbb{R}^{m \times l}$; m and l are hyper-parameters that are the maximum length of the response and knowledge sentence.

A_{rk} reflects the relationship between the response terms and the knowledge terms: for each response term, A_{rk} includes attention scores with all knowledge terms. Similarly, A_{sk} includes attention scores between SRVs and the knowledge representations. The knowledge terms with larger response-knowledge attention scores are expected to produce outputs closer to the true response. In the training phase, that is guided by the *filtering loss* for A_{rk}:

$$\mathcal{L}_{filter} = BCE(GT_{know}, Mean(A_{rk})) \tag{7.6}$$

where GT_{know} is the *knowledge ground truth* vector which indicates whether the knowledge terms appear in the corresponding response or not and BCE is the Binary Cross Entropy loss function. $Mean(\cdot)$ computes the mean values for knowledge terms (in the matrix columns) across response terms ($Mean(\cdot) \in \mathbb{R}^{l}$).

At the same time, the objective to train MLPs to create SRVs similar to the response representations V_{resp}. In each iteration, A_{sk} to A_{rk} will be computed and compared. Then the *approximation loss* function is formulated as:

$$\mathcal{L}_{approx} = MSE(Mean(A_{rk}), Mean(A_{sk})) \tag{7.7}$$

where $MSE(\cdot)$ is the Mean Squared Error function. $Mean(\cdot)$ of A_{rk} and A_{sk} produces l-length knowledge term vectors whose values are used to characterize the importance of each knowledge term. These weights are then used to update the knowledge vector:

$$\widetilde{V}_{know} = Mean(A_k) \odot V_{know} \tag{7.8}$$

where $\odot$ denotes element-wise multiplication and A_k corresponds to A_{rk} in the training phase and to A_{sk} in the test phase. V_{post} and the weighted knowledge vector $\widetilde{V}_{know}$ become input for the KTWM decoder.

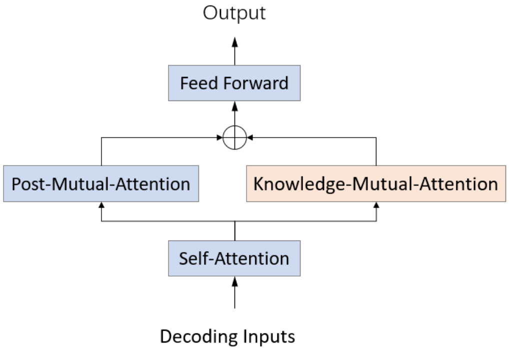


Figure 7.3: *Knowledge Term Weighting Model Decoder.*

7.1.3 KTWM Decoder

In order to incorporate multiple sources of input, a KTWM decoder is devised and it is similar to the TED model by Zheng and Zhou (2019). Figure 7.3 shows the architecture of the KTWM decoder. The blue frames are the standard Transformer decoder set-up with a self-attention layer and a mutual-attention layer (for the post), followed by a feed-forward layer.

KTWM includes an additional knowledge-mutual-attention layer that applies the same process to the knowledge, i.e., replicates the post-mutual-attention layer for the knowledge. However, while TED focuses on assigning different weights to different sources, KTWM is already provided with scored knowledge terms. V_{PMA} is used to denote the post-mutual attention, V_{KMA} for knowledge-mutual attention, and V_{dec} for the decoding tokens representation matrix. With the attention defined by Eq. (2.1), it can be expressed as:

$$V_{PMA} = Attention(V_{dec}, V_{post}, V_{post}) \tag{7.9}$$

$$V_{KMA} = Attention(V_{dec}, \widetilde{V}_{know}, \widetilde{V}_{know}). \tag{7.10}$$

The *final mutual attention* V_{MA} in the decoder is then calculated from V_{PMA} and V_{KMA}:

$$V_{MA} = V_{PMA} \oplus V_{KMA} \tag{7.11}$$

where $\oplus$ means element-wise summation. The feed-forward layer is a standard Transformer transition layer (Vaswani et al. (2017)).

Finally, Negative Log Likelihood (NLL) is adopted to train the model:

$$\mathcal{L}_{NLL} = -\sum_{t=1}^{m} logP(y_t|y_{<t}, P, k). \tag{7.12}$$

Given a post (P), knowledge (k), and the previously predicted terms ($y_{<t}$), $\mathcal{L}_{NLL}$ maximises the probability of the currently predicted term. During the training phase, $P(y_t|y_{<t}, P, k)$ is replaced with $P(r_t|r_{<t}, P, k)$, i.e., the ground truth response is used as the input instead of the model output from the previous steps (Goyal et al., 2016).

It is assumed that all three loss functions are equally important in this work and create the final loss function as a sum:

$$\mathcal{L} = \mathcal{L}_{filter} + \mathcal{L}_{approx} + \mathcal{L}_{NLL} \tag{7.13}$$

KTWM thus provides a flexible learning framework, enabling the injection of knowledge based on different selection criteria. KTWM's effectiveness is examined when it is combined with Post-KS, SKT, and TED models by incorporating the knowledge that each of these methods selects.

7.2 Experiment

7.2.1 Datasets

In the experiments, two publicly available datasets: Wizard of Wikipedia (WoW) (Dinan et al., 2019) and Holl-E (Moghe et al., 2018) are used. They are widely used in previous research, including Lian et al. (2019); Kim et al. (2020); Zheng and Zhou (2019). Both are purposefully created by human editors to support dialogue generation research. The WoW dataset includes two test sets: the seen test set and the unseen test set. These sets are categorized based on whether the topics are present in both the training set and the test set or if the topics are unseen in the training set. The details of these two datasets are introduced in Sec. 2.3.

7.2.2 Implementation Details

For comparison, a set of parameters is fixed across all the experiments. The number of dimensions in embeddings is set to be 100, and the vocabulary size is 30,000. This vocabulary is obtained by ranking terms based on word frequency in the training set. The minimum sequence length is set to 8, and the maximum length is 30. Training is done using mini-batches of size 64, and the Adam optimizer is employed for optimization (Kingma and Ba, 2015). The initial learning rate is set to 0.001 and is halved when the loss score does not decrease for two epochs. During the training phase, response-retrieved knowledge is used, meaning the sentences retrieved by the BM25 algorithm using responses as queries (see Figure 7.2). Specifically, the top 1 ranked knowledge sentence is injected into KTWM. In the test phase, knowledge is retrieved using the BM25 algorithm, with posts serving as queries. All experiments are conducted on a single TITAN V GPU. For the WoW dataset, an experiment takes about 6 hours to complete, while for Holl-E, it takes about 2.5 hours.

7.2.3 Metrics

For performance evaluation, standard lexical-based metrics, such as BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005), as well as an embedding-based metric, BOW Embedding (Liu et al., 2016b) are used. BLEU metrics measure the co-occurrence of n-gram terms in two given sequences, while METEOR is an adaptation of BLEU that considers the presence of synonyms and common word stems to better capture semantic similarity. BOW Embedding, on the other hand, measures the similarity of two sentences from the semantic perspective by comparing the Bag-of-Words representation of the sentences in the pre-trained word embedding space. Detailed introductions and discussions about these metrics are provided in Section 2.2. By employing a combination of lexical-based and embedding-based evaluation metrics, the goal is to obtain a comprehensive understanding of the performance of the proposed dialogue generation models. This combination allows us to assess not only the word-level and phrase-level similarities between the generated responses and ground truth references but also the semantic similarities between them, providing a more complete evaluation of the model's ability to generate coherent, contextually appropriate, and informative responses in dialogue tasks.

7.2.4 Baselines.

The KTWM is compared with three strong baselines:

Post-KS (Lian et al., 2019) uses an elaborate knowledge selection module and injects the selected knowledge into a generative model by approximating prior-distribution (i.e., $p(k|P)$) with posterior-distribution (i.e., $p(k|P, R)$).

SKT (Kim et al., 2020) considers knowledge selection as a sequential problem. It jointly trains a knowledge selection and a generative model by taking into account inputs and knowledge from previous turns.

TED (Zheng and Zhou, 2019) uses a knowledge-grounded generative model that assigns different weights to different sources when generating responses. It applies knowledge ranking using BM25, which is the same as that in this work.

7.2.5 Experiment Design

The experiments focus on the term weighting of the selected knowledge rather than the knowledge selection itself. Since the baseline models (Post-KS[1], SKT[2] and TED[3]) incorporate knowledge selections, a comparative evaluation of KTWM is operated with incorporating knowledge specific to each baseline method. Furthermore, since all three baselines inject knowledge at the sentence level, by selecting the top-ranked sentences, and the same to the KTWM.

7.3 Experimental Results and Analysis

7.3.1 Performance of Generating Response

The KTWM experiments with the WoW and the Holl-E datasets are summarized in Table 7.2. Results for the WoW seen and unseen test sets are in Table 7.2, sections (a) and (b), respectively. Results for the Holl-E dataset are in Table 7.2, section (c). Since METEOR extends BLEU metrics by considering word stems and synonyms, it is taken as the main metric for discussing the experiment results. We can observe that:

(1) For all three datasets, KTWM outperforms each baseline method across all lexical and embedding-based metrics with a statistically significant difference.

(2) KTWM with Post-KS knowledge achieves the largest relative improve-

[1] https://github.com/bzantium/Posterior-Knowledge-Selection

[2] https://github.com/bckim92/sequential-knowledge-transformer

[3] https://github.com/tonywenuon/Transformer_ED

Table 7.2: KTWM performance on the WoW seen and unseen test data and Holl-E dataset with different knowledge sources. Comparison with Post-KS, SKT, and TED models. '*' indicates statistical significance ($p < 0.05$). **Bold** indicates the best performance for a given metric. 'w' denotes 'with', i.e., injecting the knowledge source that is used in a specific baseline model.

Generation Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	Average	Greedy	Extrema
(a) WoW seen test data								
Post-KS	17.56	6.35	2.68	1.35	5.96	0.611	0.364	0.334
KTWM w Post-KS knowledge	21.98*	**10.03***	**5.56***	**3.44***	**8.66***	**0.684***	0.394*	**0.376***
SKT	16.45	7.97	4.75	3.14	7.29	0.639	0.385	0.366
KTWM w SKT knowledge	**22.00***	10.00*	5.47*	3.35	8.59*	0.681*	**0.398***	0.370
TED	20.26	9.43	5.32	3.35	8.45	0.658	0.385	0.366
KTWM w TED knowledge	21.86	10.02	5.51	3.35	**8.66***	0.682*	0.394*	0.374*
(b) WoW unseen test data								
Post-KS	17.25	5.58	2.03	0.81	5.50	0.598	0.352	0.305
KTWM w Post-KS knowledge	**21.66***	**8.98***	**4.41***	**2.41***	**8.50***	**0.681***	**0.388***	**0.361***
SKT	14.09	5.72	2.89	1.72	5.80	0.591	0.36	0.304
KTWM w SKT knowledge	20.46*	8.07*	3.85*	2.03*	7.77*	0.664*	0.38*	0.337*
TED	19.28	7.83	3.83	2.09	7.02	0.634	0.363	0.327
KTWM w TED knowledge	20.46*	8.32*	4.03*	2.17*	7.92*	0.668*	0.379*	0.342*
(c) Holl-E dataset								
Post-KS	14.07	7.07	4.96	3.81	5.98	0.639	0.382	0.333
KTWM w Post-KS knowledge	19.91*	11.00*	8.02*	6.42*	8.37*	0.675*	0.387*	0.350*
SKT	21.54	13.81	10.94	9.17	8.48	0.637	0.391	0.333
KTWM w SKT knowledge	**23.05***	13.96*	10.66	8.71	9.73*	0.673*	0.389	0.362*
TED	21.62	13.71	10.83	9.17	9.13	0.685	**0.414**	**0.366**
KTWM w TED knowledge	22.42*	**14.01***	**10.98**	**9.28**	**10.20***	**0.688***	0.402*	**0.366**

ment considering the METEOR score: increase of 45.3%, 54.5%, and 40.0% for the three test sets, respectively.

(3) For the Holl-E dataset, KTWM with TED knowledge outperforms the other two baseline models. TED knowledge comprises top sentences retrieved using the BM25 algorithm. However, KTWM + Post-KS knowledge works better on WoW. I attribute this phenomenon to the intrinsic difference between the two datasets because the data distributions are various across them. The dataset investigation is out of the scope of this work.

(4) On the WoW datasets, KTWM achieves a remarkable performance in terms of BLEU-1 and METEOR scores. A consistent and strong performance in the WoW unseen test data indicates the robustness and generalization of KTWM.

7.3.2 Results of Knowledge Term Weighting

The loss function (Eq. (7.6)) controls KTWM's ability to distinguish between relevant and non-relevant knowledge terms, similar to a binary classifier. A threshold of 0.5 is set for a knowledge term's predicted score and consider the

Table 7.3: Precision, Recall, and F-1 scores for the useful and noisy term predictions on the WoW seen test set.

Name	Prec	Rec	F-1
Useful Term Prediction	0.50	0.32	0.39
Noisy Term Prediction	0.92	0.96	0.94

overlap between the predicted and the truly useful knowledge terms. This leads to precision/recall evaluation of the positive and the negative class prediction. Table 7.3 shows results from the WoW seen test set. They are representative of the results for the other two datasets.

We can observe that the precision of predicting useful terms is 50% and noisy terms are over 91% (with a high F-1 score, 94%). Thus KTWM term weighting is effective in detecting noisy terms while only half of the predicted useful terms overlap with the ground truth terms. Since noisy terms are assigned lower term weights, KTWM is effectively improving the dialogue generation performance. In Sec. 7.3.4, the illustrations of the KTWM noise reduction will be presented.

7.3.3 Analysis of Input Sequence Length

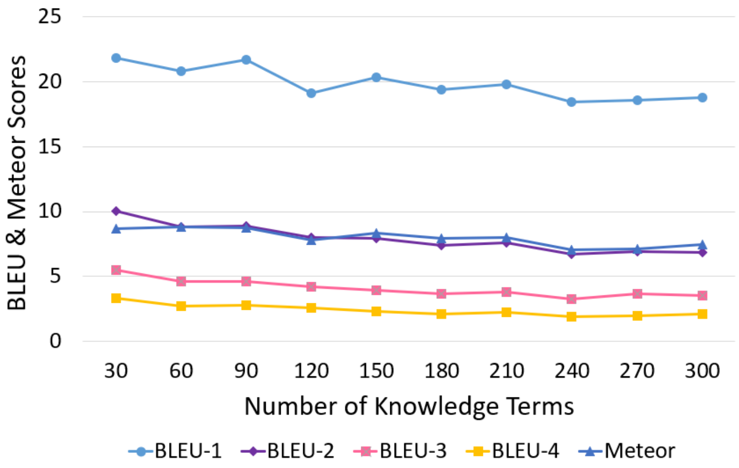


Figure 7.4: *Effects of the increased number of knowledge terms on the KTWM performance (WoW seen test set).*

I analyze the effects of knowledge de-noising by considering the *useful words proportion* (UWP) as increasing the number of injected knowledge terms: $UWP = \frac{Number\ of\ distinct\ useful\ terms}{Number\ of all\ injected\ words}$. UWP_N is used as an instantiation for UWP when N knowledge terms are injected (e.g., UWP_{30} for 30 knowledge terms). The analysis shows that UWP_{30} is 12.23% and UWP gradually decreases with additionally injected knowledge leading to UWP_{300} of only 3.35%. Figure 7.4 shows a gradual decline of the KTWM performance with the increased length of injected knowledge, as the proportion of noisy terms increases.

The effects of the loss functions $\mathcal{L}_{filter}$ and $\mathcal{L}_{approx}$ are also investigated on the KTWM performance by running experiments with and without them. In Table 7.4 the results are shown on the WoW seen test set using BM25 to select knowledge. Note that, after removing $\mathcal{L}_{filter}$ loss function, BLEU-1 and Average scores decrease, while BLEU-4 and METEOR scores increase. Since $\mathcal{L}_{filter}$ aims to ensure that relevant response terms are promoted, it is not surprising that the metrics focused on unigrams are most affected. However, this impact on KTWM is less notable than the removal of the $\mathcal{L}_{approx}$. Without $\mathcal{L}_{approx}$, the KTWM loses the ability to align simulated response vectors SRVs with the response embeddings to capture the attention distribution between the knowledge and the response embeddings that are needed to score knowledge terms. This increases the noise ratio and reduces the KTWM performance scores across all metrics.

Table 7.4: Ablation study of the multi-component loss function on the WoW seen test set. w/o means ‘without’.

Name	BLEU-1	BLEU-4	METEOR	Average
KTWM	21.86	3.35	8.66	0.682
w/o $\mathcal{L}_{Filter}$	20.69	3.67	8.77	0.661
w/o $\mathcal{L}_{Approx}$	7.49	1.59	5.42	0.598

7.3.4 Good cases and bad cases of Knowledge Term Weights and KTWM Generated Responses

In Table 7.5 and 7.6, examples of post/response pairs and selected knowledge with terms weighted by KTWM are given. As explained in Sec. 7.3.2, a threshold of 0.5 is used on term scores to classify terms into useful and noisy ones and study the effect of this selection on the overall performance of KTWM. In the examples, the weights of each term are visually shown. Terms are highlighted in different shades of blue color according to the weight (note the color legend at the bottom of the tables). All the examples are extracted from the WoW seen test set. They are sorted by the number of words that exceed the threshold.

In Table 7.5, we can see that the keywords are tagged with dark blue, indicating that KTWM has assigned high weights to them. From the KTWM-generated responses, it can be seen that if the words appear in the post and ground-truth response simultaneously, the KTWM works effectively, i.e., can correctly incorporate injected knowledge into the generated response. On the other hand, the negative examples in Table 7.6 show that the term scoring can be ineffective if there is no good overlap with the ground truth response. We can observe in these examples that most of the words with relatively high scores do not exist in both post and response. At the same time, if the injected knowledge does not contain useful terms, the produced responses might be irrelevant. In Table 7.6, most of the terms have a light blue color, indicating that KTWM detected the relatively low importance of these terms correctly. The examples in these two tables also confirm the statistical results shown and discussed in Sec. 7.3.2. KTWM term weights still induce noise, especially when the injected knowledge does not contain useful terms (i.e. terms that are present in the ground truth response), resulting in a worse response generation performance. Note that both sets of examples include highlighted punctuation (e.g., ‘,’) and language structural terms (e.g., ‘the’, ‘is’) which obtain high KTWM weights. I assume that such terms are widely distributed in post and response sets and therefore detected as important.

7.4 Summary

Dialogue models that are based on knowledge use either traditional unsupervised retrieval techniques like BM25 or incorporate knowledge selection into the generation model itself. In the majority of cases, knowledge is incorporated as whole sentences or paragraphs. Prior research (Galley et al. (2019), Zheng et al. (2020)) has demonstrated that incorporating useful terms can improve response generation performance, but it is important to mitigate the negative effects of noisy terms.

In this research, a novel Knowledge Term Weighting Model (KTWM) is proposed for weighting and de-noising knowledge terms at the term level. KTWM demonstrates the effective estimation of knowledge term weights and outperforms state-of-the-art baseline models in terms of response generation performance, as demonstrated by evaluations on two widely used datasets.

Additionally, valuable insights into the significance of knowledge term weighting are offered by this research. As part of the future work, there are plans to (1) broaden the KTWM models to encompass multiple sources of evidence, such as striking a balance between selected knowledge and dialogue contexts (i.e., prior dialogue turns), and (2) leverage pre-trained models as the backbone due to their superior performance. (3) Furthermore, the aim is to construct a unified mechanism that denoises both context and knowledge at the sentence level and term level.

Table 7.5: Positive cases that the injected knowledge is properly weighted by the proposed KTWM model. The background color stands for the probability of being a useful term.

Post: I love avengers!! Have you watched the Infinity War movie? I think it was really good and I can't wait till the next movie to find out what happens. **Response:** The Avengers Infinity War was so good. What a cliff hanger. Blows my mind that the Avengers made their comic debut way back in 1963. **Knowledge Terms Weighted by KTWM:** In ” Avengers : Infinity War ” , the Avengers join forces with the guardians of the galaxy to confront Thanos , who is trying to collect the infinity stones **Response generated by KTWM:** I love the Avengers. the Avengers was the Avengers, Infinity War. He is such a great show.
Post: I have never actually been to a horse race in Kentucky, where is the Derby held at anyways? **Response:** Louisville Kentucky, it ends the Kentucky Derby festival usually. **Knowledge Terms Weighted by KTWM:** The Kentucky Derby presented by Woodford reserve is a horse race that is held annually in Louisville , Kentucky , United States , on the first Saturday in may . **Response generated by KTWM:** I'm not sure but I know that it is located in Kentucky race in the United States and Canada.
Post: New York city is quite impressive. **Response:** Yeah they say that New York and LA are the two largest markets in our country. **Knowledge Terms Weighted by KTWM:** The city of New York , often called New York City or simply New York , is the most populous city in the United States **Response generated by KTWM:** Yes, it is. The city of New York is the most populous city in the United States.
Post: Hi there! I drive a Honda Civic. Do you have a car? **Response:** I have a VW golf, but was looking to change to a Honda Civic. I heard that the Civic has got larger throughout the years. **Knowledge Terms Weighted by KTWM:** The Honda Civic type R is the highest performance version of the Honda Civic made by Honda Motor company of Japan **Response generated by KTWM:** I have a Honda Civic. They are the second most commonly used in the US in Japan.
Post: Well that's interesting, but seems like it could get confusing! I bet humans have been hiking ever since we could walk though! **Response:** Yeah hiking has been around for ever, the term bush walking is endemic to Australia. **Knowledge Terms Weighted by KTWM:** Hiking is the preferred term , in canada and the united states , for a long , vigorous walk , usually on trails (footpaths) , in the countryside **Response generated by KTWM:** Well, hiking is a long term hiking is used in the United States, and the countryside on trails.

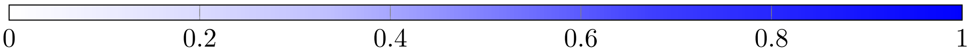
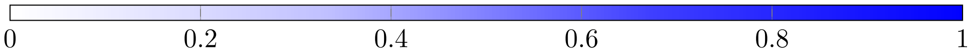


Table 7.6: Negative cases that the injected knowledge is not properly weighted by the proposed KTWM model. The background color stands for the probability of being a useful term.

Post: Yes, indeed. What cultural meanings come to mind when you think of dance? **Response:** Some of its many functions that come to mind are social, ceremonial, and competitive in meaning. **Knowledge Terms Weighted by KTWM:** One open question regarding the nature of the mind is the mind body problem , which investigates the relation of the mind to the physical brain and nervous system **Response generated by KTWM:** I'm not sure but I do know that the spiritual core is associated with mind and mind.
Post: I have written a few historical fiction novels that I am currently trying to get published. **Response:** That's really cool, historical fiction is such a fun genre to read. **Knowledge Terms Weighted by KTWM:** Historical romance (also historical novel) is a broad category of fiction in which the plot takes place in a setting located in the past **Response generated by KTWM:** That's great. I love romance novels. It's a broad category of historical fiction, and fantasy.
Post: I would have to say Snooki still, she is such a ball of fire. **Response:** Since you like Snooki so much, did you happen to see the Spinoff of Jersey shore called ”Snooki and Jwoww”? **Knowledge Terms Weighted by KTWM:** She attended Columbia high school , in east greenbush , New York , and New York Institute of Technology **Response generated by KTWM:** I don't know much about Vietnamese style, but I know it has been around since 1700
Post: Are they popular in other areas of the world like Europe or Asia ? **Response:** Similarly! In the European Union vehicles in this size are known as large good vehicles. **Knowledge Terms Weighted by KTWM:** Western civilisation traces its roots back to western Europe and the western Mediterranean **Response generated by KTWM:** I'm not sure, but I do know that they are native to the Mediterranean Mediterranean cuisine.
Post: Wow that's interesting. The food world has really diversified. **Response:** Yeah or you can cook without heat. Like in south America they make Ceviche, which is fish that is cooked in lemon or lime juice and the acid cooks the fish. **Knowledge Terms Weighted by KTWM:** Bow Wow Wow are an English 1980s new wave band , created by Malcolm Mclaren **Response generated by KTWM:** Yes, it is! It's the world's most popular foods in the world.

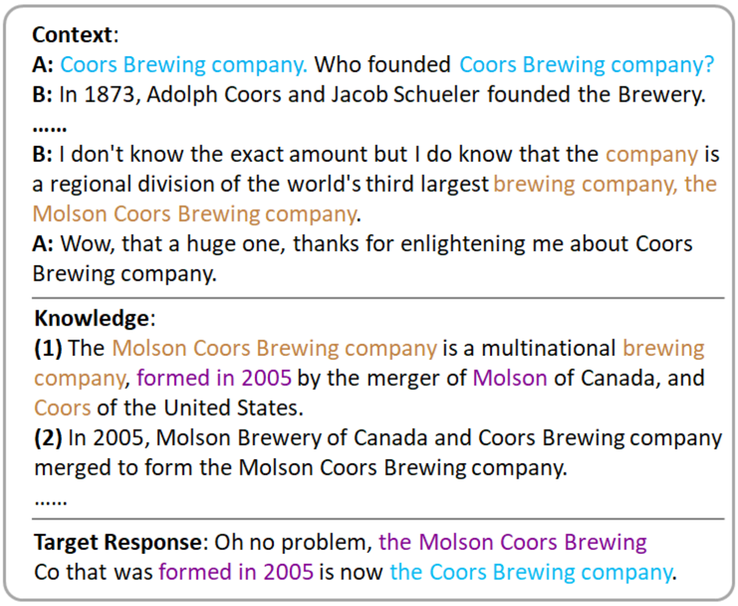


Chapter 8

A Unified Dialogue Generation Model with Contextual Knowledge Learning

The preceding Chapters have explored several critical research questions that pave the way for constructing a unified dialogue generation model that leverages both context and knowledge. Specifically, the follows have been discussed: (1) the usage of context alone to generate responses; (2) the integration of knowledge into a generative model; (3) the selection of pertinent knowledge at the sentence level; and (4) the implementation of term level de-noising for the input knowledge sequences.

In this Chapter, the method of weighing both context and knowledge will be further discussed to reach a unified dialogue generation framework. The research community has been actively exploring the incorporation of external knowledge resources and conversational history as the context for response generation. For example, Ghazvininejad et al. (2018) examined a knowledge-grounded dialogue generation method using the multi-task learning paradigm. Additionally, Zheng and Zhou (2019) projected knowledge sentences into a generative model by assigning varying weights to knowledge, and Kim et al. (2020) treated knowledge selection as a sequential decision problem. Recently, the combination of knowledge selection and dialogue generation based on pre-trained models has become possible, as proposed by Liu et al. (2021c) and Prabhumoye et al. (2021).



Context:
A: Coors Brewing company. Who founded Coors Brewing company?
B: In 1873, Adolph Coors and Jacob Schueler founded the Brewery.
……
B: I don't know the exact amount but I do know that the company is a regional division of the world's third largest brewing company, the Molson Coors Brewing company.
A: Wow, that a huge one, thanks for enlightening me about Coors Brewing company.

Knowledge:
(1) The Molson Coors Brewing company is a multinational brewing company, formed in 2005 by the merger of Molson of Canada, and Coors of the United States.
(2) In 2005, Molson Brewery of Canada and Coors Brewing company merged to form the Molson Coors Brewing company.
……

Target Response: Oh no problem, the Molson Coors Brewing Co that was formed in 2005 is now the Coors Brewing company.

Figure 8.1: *Example showing utterances of participants A and B, the scored knowledge, and the target response. The knowledge sentence (1) is deemed the best knowledge for response generation by the proposed CKL model. The best context segments for retrieving the best knowledge are colored Brown; the best context segments for generating response are colored Blue and the best knowledge segments for the response are Purple.*

However, it has been shown that adding knowledge indiscriminately, can hurt performance. Thus, the context has been used to select the best knowledge for the response generation. Since the context itself consists of multiple utterances, the same concern applies: not all the prior utterances are equally useful for generating the response. Therefore, the context needs to be evaluated for its importance in relation to generating the response and identifying the relevant knowledge, separately. In Figure 8.1, an example is given from the Wizard of Wikipedia (test seen) dataset, illustrating that the best context segments for response generation may not necessarily be the best context segments for retrieving the best knowledge. Furthermore, both the context and the contextual knowledge contribute to the coverage of the target response (blue and purple words). Thus, it is important to devise effective learning methods to identify the best context for response generation and knowledge selection. Once the knowledge is selected, there is still a question of whether and how to refine its selection for optimal use.

Recent studies differentiated between the two context roles by adopting a pipeline approach and training different models for each of them. Zheng et al. (2020) proposed a knowledge retrieval model TPPA that re-orders retrieved knowledge guided by its relevance to the response and investigated the effects of the resulting knowledge sets in combination with generative models such as TED (Zheng and Zhou, 2019) and WSeq (Tian et al., 2017). Paranjape et al. (2021) introduced a posterior-guided training to guide the retrieval of the relevant knowledge. A BART-based generative model (a generator) is used to generate responses but the retriever and the generator are trained independently. Similarly, Glass et al. (2022) developed a Re^2G model which comprises a retriever, a re-ranker, and a generator. The re-ranker can take as input the outputs of multiple retrieval systems, e.g., ANN-based retrieval and BM25 method and the content retrieval training is an integral part of the content generation. This approach can differentiate the context roles from knowledge selection and response generation tasks. However, it requires additional training stages, which may incur and accumulate additional errors, and cannot separate the context information used for knowledge selection and response generation within the unified model.

In this work, a hypothesize is made that the integrated approach to model training and selection of context and knowledge can be improved through a parallel learning architecture where specific content selection roles (context and knowledge) are clearly differentiated and each learning facet is supervised, controlling for model training. Guided by the hypothesis, a Contextual Knowledge Learning (**CKL**) model is proposed, in which *Latent Vectors* are designed to capture context roles and knowledge characteristics: the *Context Latent Vector* for the relationship of context to the responses and to the 'best' knowledge, and the *Knowledge Latent Vector* for the knowledge to capture the importance of knowledge to the responses. *Latent Weights* are then derived from the Latent Vectors to indicate the importance of context utterances and knowledge sentences.

I also extend the notion of the Attention operation, where tokens' attention scores are entirely decided by the scaled dot product between two representations, and devise a *Latent Weight Enhanced Attention*. The attention operation is augmented with the multiplication by the tokens' attention scores and the *Latent Weights* (i.e., the context utterance's weight and knowledge sentence weight). By adopting the weak supervision technique, the *Latent Weights* for context and knowledge are supervised by the (noisy) pseudo ground truth, removing the need for human annotations. Combined with the Negative Log Likelihood loss, the CKL is trained in a unified way, differentiating the context utterances for the knowledge selection and response generation tasks.

The performance that the CKL model obtains is superior to six strong baseline approaches, including Transformer-based and pre-trained model-based methods, on two publicly available datasets Wizard of Wikipedia and CMDoG. By experimenting with a 50% smaller training set, the CKL still outperforms the baseline methods.

8.1 Methodology

Overall, the CKL model (Figure 8.2) consists of four components: an encoder, a Context Latent Weight generator (*CLW* Generator), a Knowledge Latent Weight generator (*KLW* Generator), and a decoder. The state-of-the-art Transformer-based encoder-decoder model BART (Lewis et al., 2020) is adopted as the backbone of the Encoder and Decoder. The *CLW* generator takes responsibility for producing two sets of context latent weights, one set for response generation (*CLWR*) and another set for knowledge latent weight generation (*CLWK*). Similar to the *CLW* generator, the *KLW* generator is used

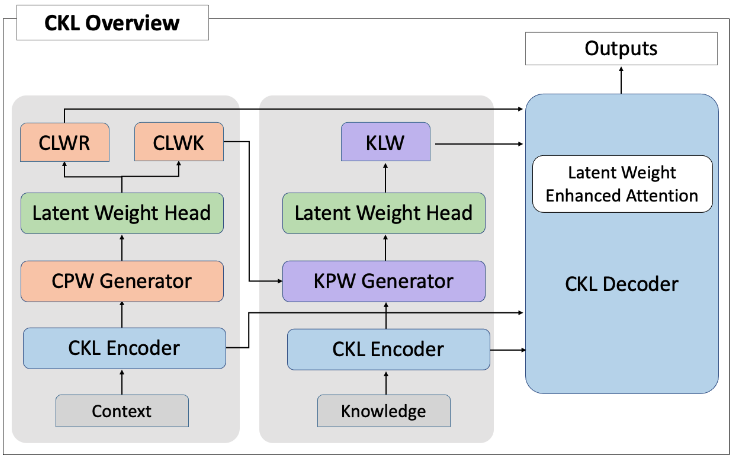


Figure 8.2: *Overview of the Contextual Knowledge Learning (CKL) model.*

to generate knowledge latent weights (KLW) which are conditioned on the context and knowledge. Finally, the decoder is a normal BART decoder but equipped with the latent weight-enhanced attention mechanism. The detailed illustration is shown in Figure 8.3 and is introduced as follows.

Problems and Definitions. Considering a conversational history that comprises context $C = \{c_1, c_2, \ldots, c_m\}$, the goal is to generate a response $R = \{r_1, r_2, \ldots, r_L\}$ by leveraging knowledge $K = \{k_1, k_2, \ldots, k_l\}$ that is relevant to the context C. Among the notations, r_i is each word of the response, c_i means the context utterance, and k_i denotes the knowledge sentence. L is the maximum token number of the response; m is the number of context utterances and l is the number of background knowledge sentences.

The aim is to (1) calculate latent weights of context utterances and knowledge sentences (Sec. 8.1.2 and 8.1.3); (2) generate the final response given context, knowledge, and their latent weights (Sec. 8.1.4). In the proposed approach, latent vectors are not integrated into the content representation. Instead, they are transformed into scalar values, referred to as *latent weights*. The Context Latent Weights for Response and Knowledge, ($CLWR$) and ($CLWK$), respectively, are used in the loss function and by the decoder to score content utterances. The Contextual Knowledge Latent Weights (KLW) are similarly used in the loss function and the decoder to score knowledge sentences.

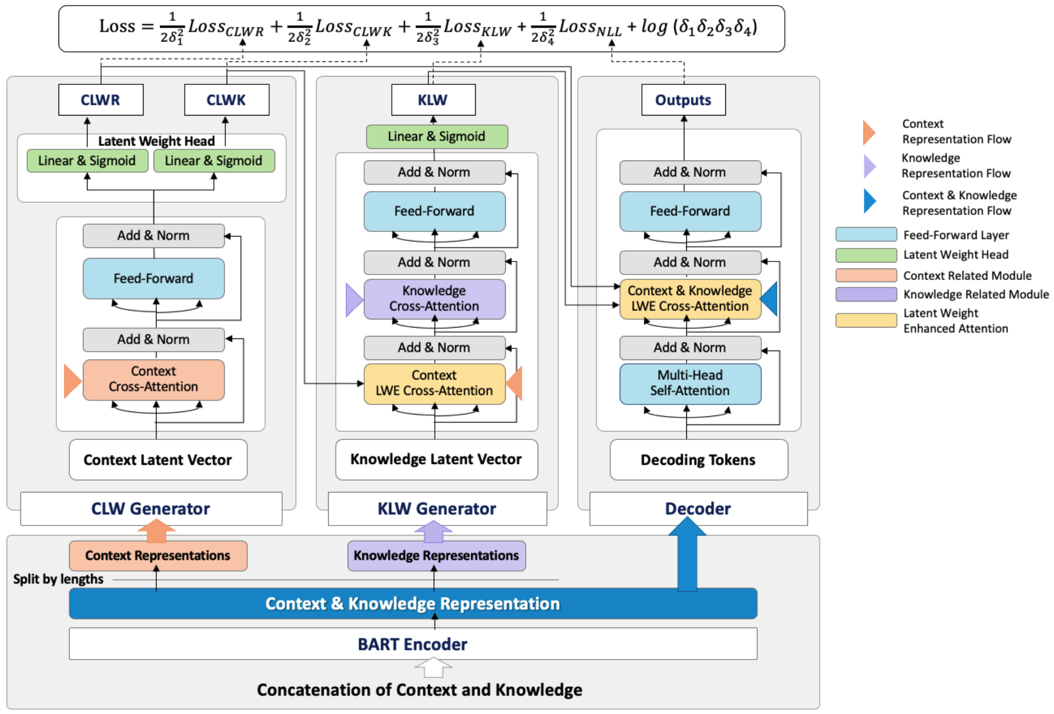


Figure 8.3: *Contextual Knowledge Learning (CKL) model architecture comprising the BART-based encoder, two generators for latent vectors training and latent weight generation, and decoder with context and knowledge scoring. Trainable parameters δ_i balance multi-losses.*

8.1.1 Encoder

Leveraging the pre-trained model BART, the BART encoder is directly used to get the context and knowledge representations. The proposed CKL model needs context utterances' and knowledge sentences' representations, so they are expected to be passed through the BART encoder sequence by sequence. However, that would destroy the inner dependency between words from sequences, i.e., this means discarding the long dependency between context and knowledge. To tackle this, firstly, the concatenation of the context and knowledge is injected to get a whole sequence representation, i.e., 'Context & Knowledge Representation' in Figure 8.3. Then by recognizing the context utterances' lengths and knowledge sentences' lengths, the whole representation is split into several sub-sequences, obtaining representations that take word long-dependency into account.

8.1.2 Context Latent Weight Generator

As shown in Figure 8.3, *CLW* Generator is designed to generate two sets of context latent weights: Context Latent Weight for Response (*CLWR*) and Context Latent Weight for Knowledge (*CLWK*).

Context Latent Vector. The *CLW* generator starts from a Context Latent Vector which is a trainable vector. Practically, it is a word embedding indexed by a fixed word index of 1.

Context Latent Vector Interaction with Context Representations. Here in the *CLW* generator, like the Transformer architecture, a standard cross-attention, feed-forward, and residual network are used. The cross-attention operation will be explained in detail because it will be used in the next sections. For the rest of the Transformer modules, e.g., feed-forward layer and residual network, please refer to Vaswani et al. (2017). Formally, the attention is calculated as:

$$Attention(Q,\ K,\ V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{8.1}$$

in which Q, K, and V are matrices and d is the representation dimension. Through the softmax function, the attention weights, i.e., QK^T, are normalized. Multiplying with V, the Q's representation is updated by K and V. In the *CLW* generator, Q is the context latent vector, while K and V are the context representation.

Latent Weight Head. The Latent Weight Head module contains a linear layer and a Sigmoid function. The purpose is to transfer the d dimensional context latent vector to scalar values. By doing so, each context utterance will have a latent score. To be specific, the Latent Weight Head is defined as follows:

$$CLWR = Sigmoid(x_{CLV}W_1 + b_1) \tag{8.2}$$

$$CLWK = Sigmoid(x_{CLV}W_2 + b_2) \tag{8.3}$$

where $CLWR \in \mathcal{R}^{1\times m}$ and $CLWK \in \mathcal{R}^{1\times m}$ are the context latent weight scores for the response and knowledge respectively. W_1, $W_2 \in \mathcal{R}^{d\times 1}$ and $b_1, b_2 \in \mathcal{R}^1$ are trainable parameters. $x_{CLV} \in \mathcal{R}^{1\times d}$ denotes the vector converted from the Context Latent Vector. It is important to note that $CLWR$ and $CLWK$ have the same Latent Weight Head architecture, but do not share parameters. $CLWR$ is used to identify the importance of a context utterance when generating responses and $CLWK$ is used when producing knowledge latent weights, i.e., knowledge sentences' importance.

Weak Supervision on *CLWR* and *CLWK*. As illustrated, when predicting the response and producing the knowledge latent weight, the role of the context utterances should not be treated as the same: $CLWR$ and $CLWK$ reflect the difference. Two loss functions are devised to weakly supervise them. The latent weight scores are expected to be a continuous value thus this task is viewed as a regression task rather than a classification task. Mean Squared Error (MSE) is adopted as the loss function. To obtain the pseudo-ground truth, the F1 score is used as a measurement of the closeness between a context utterance and the response on the word level. As for its values, for $CLWR$, the context utterance with the maximum F1 score is tagged as 1 and the rest of the utterances to be 0. It is worth noting that the last utterance, i.e., the post, is always 1 because it has been proven crucial for response generation Sankar et al. (2019b).

For training $CLWK$, the method is the same for constructing the pseudo ground truth. The only difference is that $CLWK$ is built for the knowledge latent weight, so the most relevant knowledge is produced for the F1 score calculation. First of all, the TF-IDF approach is used to retrieve from the knowledge sentences by taking the response as the query, i.e., ranking the knowledge sentence by *TF-IDF*(knowledge sentence, response).[1] The top-1 ranked sentence based on the TF-IDF is treated as the most important knowledge sentence, being tagged as *Top1-RK*. Secondly, similar to $CLWR$,

[1] TF-IDF($\cdot$) is the TF-IDF (term frequency-inverse document frequency) function. IDF is obtained by the individual dataset.

the context utterance with the maximum F1(Context Utterance, *Top1-RK*) is used to supervise *CLWK*. Formally,

$$GT_{CLWR(i)} = \begin{cases} \text{1, if c}_i\text{= argmax(F1(c}_i\text{, R))} \\ \text{1, if c}_i\text{= post} \\ \text{0, otherwise} \end{cases} \tag{8.4}$$

$$GT_{CLWK(i)} = \begin{cases} \text{1, if c}_i\text{= argmax(F1(c}_i\text{, } \textit{Top1-RK))} \\ \text{1, if c}_i\text{= post} \\ \text{0, otherwise} \end{cases} \tag{8.5}$$

in which c_i means each context utterance. Then, the loss function is defined to be:

$$Loss_{CLWR} = MSE(CLWR, GT_{CLWR}) \tag{8.6}$$

$$Loss_{CLWK} = MSE(CLWK, GT_{CLWK}) \tag{8.7}$$

where GT_{CLWR} and GT_{CLWK} are the pseudo-ground-truth context utterance scores for response generation and knowledge selection tasks respectively.

8.1.3 Knowledge Latent Weight Generator

The knowledge Latent Weight generator is designed to generate a knowledge latent weight (KLW). It begins with a knowledge latent vector, which is a word embedding indexed by a fixed index of 1. Note that the knowledge latent word embedding is different from the context latent embedding.

Latent Weight Enhanced Attention. Latent Weight Enhanced Attention (LWE Attention) is built on top of the standard attention by considering the latent weights. Originally, the attention is calculated between two sequence representations from the word level (shown in Eq. 8.1). The LWE Attention takes sentence-level scores, i.e., the latent weights, into consideration. By this, the Eq. 8.1 is then changed to be:

$$LWE\ Attention(Q,\ K,\ V) = LW \times softmax\left(\frac{QK^T}{\sqrt{d}}\right) V \tag{8.8}$$

where *LW* stands for latent weights. *LW* will be different when predicting responses and generating knowledge latent weight. Namely, in the *KLW* generator, the *LW* is replaced with *CLWK*. In the Decoder, it is changed to *CLWR* and *KLW*, which will be introduced in Sec. 8.1.4.

Context & Knowledge Dependency (CK-Dep for short). Prior studies Prabhumoye et al. (2021); Liu et al. (2021c) consider the context and knowledge dependency by stacking a context cross-attention and a knowledge cross-attention from word level. I also leverage the stacked architecture and consider the context sentence-level weights (through LWE Attention), i.e., the context LWE cross-attention module and the knowledge cross-attention module in *KLW* generator.

Weak Supervision on *KLW*. After going through the CK-Dep operation, the Latent Vector is processed by the Latent Weight Head module to get the *KLW*. The knowledge generally contains richer information than the context Zheng and Zhou (2019); Kim et al. (2020). For context, the top-1 ranked utterance is taken as the pseudo ground truth GT_{CLWK}. However, for knowledge, a hyper-parameter N is set to get the pseudo ground truth knowledge sentences GT_{KLW}. Namely, the top N ranked knowledge sentences are considered to be the ground truth for supervising *KLW*.

$$GT_{KLW(i)} = \begin{cases} 1,\ \mathrm{k}_i\text{=Top N argmax(TF-IDF(k, R))} \\ 0,\ \text{otherwise} \end{cases} \tag{8.9}$$

where k_i represents each knowledge sentence. Similar to the *CLWR* and *CLWK*, the *KLW* generation is also considered as a regression task, and the loss function is:

$$Loss_{KLW} = MSE(KLW, GT_{KLW}) \tag{8.10}$$

8.1.4 Decoder and Training

The Decoder is a BART decoder but equipped with LWE Attention. In the 'Context & Knowledge LWE Cross-Attention' module in Figure 8.3, the context and knowledge representations are multiplied by the corresponding latent weights. Namely, the LW in Eq. 8.8 will be replaced by $CLWR$ and KLW when dealing with context and knowledge in the Decoder. Formally, the Eq. 8.8 is instantiated to be:

$$\begin{aligned} &\mathit{PE\ Attention(Q,\ K,\ V)} = \\ &\sum_{i=1}^{m} CLWR_i \times \mathit{softmax}\left(\frac{QK_i^T}{\sqrt{d}}\right) V_i + \\ &\sum_{j=1}^{l} KLW_j \times \mathit{softmax}\left(\frac{QK_j^T}{\sqrt{d}}\right) V_j \end{aligned} \tag{8.11}$$

where, K_i and V_i means i-th context utterance. K_j and V_j denote j-th knowledge sentence. $CLWR_i \in \mathcal{R}^1$ and $KLW_j \in \mathcal{R}^1$ stand for the corresponding context and knowledge latent weights. The loss function for response generation is a Negative Log Likelihood loss (NLL).

$$Loss_{NLL} = -\sum_{i=1}^{L} \mathit{log}\ p(R_t|R_{<t},\ C,\ K). \tag{8.12}$$

in which, L is the maximum length of the response, t is the t-th token to be generated and $R_{<t}$ denotes the generation steps prior to t.

Aggregation of Loss Functions. In this paper, there are four different loss functions, including Eq. 8.6, Eq. 8.7, Eq. 8.10 and Eq. 8.12. Previous studies simply aggregate different loss functions by either an addition operation Li et al. (2019b); Zheng et al. (2021) or setting hyper-parameters to do a weighted sum Wu et al. (2021b), which are sub-optimal. Kendall et al. (2018) propose a principled approach to multi-task learning which weighs multiple loss functions by considering the homoscedastic uncertainty of each task. This Automatic Weighted Loss (AWL) allows the model to simultaneously learn various quantities with different units or scales in various settings. Adopting this strategy, the final loss's formulation is:

$$\begin{aligned} Loss = \frac{1}{2\delta_1^2} Loss_{CLWR} + \frac{1}{2\delta_2^2} Loss_{CLWK} + \\ \frac{1}{2\delta_3^2} Loss_{KLW} + \frac{1}{2\delta_4^2} Loss_{NLL} + log(\delta_1 \delta_2 \delta_3 \delta_4) \end{aligned} \tag{8.13}$$

The final goal is to minimize the objective with respect to $\delta_1, \delta_2, \delta_3$ and δ_4 as learning the relative weight of the four different losses.

8.2 Experiment

8.2.1 Datasets

Following previous research practices (Prabhumoye et al., 2021; Liu et al., 2021c; Li et al., 2019b; Zhao et al., 2020a), two public datasets are used for conducting the experiments: Wizard of Wikipedia (Dinan et al., 2019) and CMU-DoG (Zhou et al., 2018). Both of them are designed for knowledge-grounded dialogue generation tasks. By using these two diverse and challenging datasets, the aim is to thoroughly assess the performance of the proposed model and its ability to handle the knowledge-grounded dialogue task. The detailed dataset introductions are provided in Section 2.3.

8.2.2 Implementation Details

In the experiments, the BART-base model[2] is used. The maximum source length is 1024 tokens and 64 tokens for the target length. For the number of context utterances, the latest 10 utterances are used. In terms of knowledge, it is also decided by the maximum source length. The learning rate is set to be 5e-5. All of the experiments are trained for 10 epochs on a single TITAN V GPU. The proposed CKL model needs about 20 hours for training on the WoW dataset and about 8 hours on the CMU-DoG dataset.

[2] https://huggingface.co/facebook/bart-base

8.2.3 Metrics

Automatic Evaluation As used in previous works (Ghazvininejad et al., 2018; Zhao et al., 2020a; Li et al., 2019b; Zheng et al., 2021), BLEU (Papineni et al., 2002), Rouge (Lin, 2004), Diversity (Li et al., 2016a) and embedding-based metric, BOW Embedding (Liu et al., 2016b) are employed as the metrics. As discussed in Sec. 2.2, Liu et al. (2016a, 2021e) suggest that compared with the other metrics, BLEU-2, and embedding-based metrics have a better correlation with human assessment, and thus in this chapter, BLEU-2 and embedding-based measurements are taken as the main metrics for discussion.

Human Evaluation To have a better understanding of the proposed model, human evaluation is done by deploying users from the crowd-sourcing Amazon MTurk platform. 5 AMT workers were employed to assess samples from 4 perspectives: Relevance, Coherence, Informativeness, and Overall Preference. Following Ling et al. (2021), the four criteria are referred to as:

- **Relevance** - whether the generated response is relevant to the given context.
- **Coherence** - whether the generated response is a coherent and meaningful continuation of the dialogue.
- **Informativeness** - how many new and diverse expressions do the generated responses introduce.
- **Overall Preference** - personal preference between two responses.

For evaluation, I randomly select 100 samples from the outputs of the proposed CKL model and DIALKI (the best-performing baseline model) for both the Wizard of Wikipedia and CMU-DoG datasets. The Amazon Mechanical Turk platform is then used to gather assessments from human evaluators. The assessors are presented with pairs of responses generated by the CKL and DIALKI models and are asked to choose the response they prefer based on different perspectives, including relevance, coherence, informativeness, and overall quality. To ensure a fair evaluation, the assessors are not informed which model generated each response. Additionally, they are given the option to consider both responses as equal with respect to the given context, i.e., 'Tie'.

8.2.4 Baselines

The CKL model is compared with six baselines.

ITDD Li et al. (2019b) proposes an incremental Transformer architecture to improve context coherence and knowledge correctness.

DRD Zhao et al. (2020a) proposes a disentangled response decoder to isolate parameters that depend on knowledge-grounded dialogues from the entire generation model.

ZRKGC Li et al. (2020) treats the knowledge as latent variables so that the model can estimate the knowledge representation distribution from the latent space.[3]

DIALKI Wu et al. (2021b) proposes a knowledge identification model to provide dialogue-contextualized passage encodings and locate knowledge that is relevant to the conversation.[4]

KAT Liu et al. (2021c) devises a three-stage architecture to get better context inner-relationship, knowledge representation, and interaction between context and knowledge.[5]

DoHA Prabhumoye et al. (2021) focuses on building a context-driven representation of the document and enabling specific attention to the information in the document.[6]

[3] https://github.com/nlpxucan/ZRKGC

[4] https://github.com/ellenmellon/DIALKI

[5] https://github.com/neukg/KAT-TSLF

[6] https://github.com/shrimai/Focused-Attention-Improves-Document-Grounded-Generation

8.3 Experimental Results and Analysis

8.3.1 Dialogue Generation Results on Automatic Evaluation and Human Evaluation.

The experimental results are shown in Table 8.1. First, based on BLEU-2 and Rouge-L scores, the proposed CKL models perform consistently better than the baseline approaches. This reflects that the results from the CKL share more consecutive tokens with the ground truth responses. Looking closely at the BLEU-2 scores, the CKL's results improve by large margins compared to the best results of the baseline approaches (DIALKI); they are around 15% better for the WoW test seen (improving from 13.72% to 15.80%), about 15% improvement for WoW test unseen (from 13.96% to 16.05%) and around 14% better for the CMU-DoG dataset.

Second, for the embedding-based metrics, the CKL is better than most of the baseline models except for the ZRKGC model on the Embedding Average measurement. However, the Extrema and Greedy scores of the ZRKGC model are lower than the CKL. This means although the generated responses of the ZRKGC model are closer to the ground truth response on average, it can not semantically capture the most important words.

Third, in terms of the diversity scores, the proposed CKL does not improve over other models. There is potential for enhancement in this aspect by refining the utilization of latent weights with the range which are currently normalized between 0 and 1 and multiplied by the word attention scores. Despite CKL model's generated responses being not the most diverse among all compared models, the human evaluation results (Table 8.2 reveal that the CKL is preferred by the 5 annotators with moderate agreement, in terms of relevance, coherence, informativeness, and overall preference.

The human evaluation results are shown in Table 8.2, from which we can see that for all of the four criteria, the proposed CKL is better than the

Table 8.1: Automatic evaluation results on Wizard of Wikipedia (WoW) test seen/unseen and CMU-DoG datasets. * means significant test value with $p < 0.05$, compared to the CKL. Note that the results of ITDD and DRD are copied from the papers, so they do not have significant test results. ‘w/o’ means without a certain module for the ablation study. All values are expressed as percentages (%).

Models	Wizard of Wikipedia test seen					
	BLEU-2	Rouge-L	Div-2	Average	Extrema	Greedy
ITDD	7.10	-	-	-	-	-
DRD	11.50	-	-	-	-	-
ZRKGC	8.80*	16.86*	22.66*	**72.32***	40.40*	41.67*
KAT	9.28*	16.41*	**45.99**	67.86*	39.06*	39.37*
DoHA	11.70*	21.32*	31.47	69.91*	40.91*	41.64*
DIALKI	13.72*	22.10*	41.71	70.43*	42.31*	41.73*
CKL	**15.80**	**23.96**	36.36	71.11	42.95	42.54
w/o $Loss_{KLW}$	14.80	23.25	37.27	70.69*	42.56	42.02*
w/o $Loss_{CLWR}$	15.43	23.75	37.38	71.05	**43.50***	42.39
w/o $Loss_{CLWK}$	15.49	23.81	37.11*	70.99	43.37*	42.50
w/o *CK-Dep*	15.51	23.82	36.99*	71.18	43.35*	**42.66**

Models	Wizard of Wikipedia test unseen					
	BLEU-2	Rouge-L	Div-2	Average	Extrema	Greedy
ITDD	4.70	-	-	-	-	-
DRD	10.10	-	-	-	-	-
ZRKGC	8.50*	16.81*	15.78*	**71.93***	39.11*	41.43*
KAT	8.49*	15.60*	**31.13***	67.02*	37.20*	38.55*
DoHA	10.53*	20.04*	21.97*	68.61*	37.96*	40.23*
DIALKI	13.96*	22.02*	25.67	69.67*	40.94*	41.39*
CKL	**16.05**	**23.93**	18.60	70.36	**41.93**	**41.86**
w/o $Loss_{KLW}$	15.13	23.39	20.25	69.99*	41.51	41.62
w/o $Loss_{CLWR}$	15.48	23.53	19.26	69.97*	41.68	41.73
w/o $Loss_{CLWK}$	15.13	23.33	19.67	69.85*	41.59	41.56*
w/o *CK-Dep*	15.55	23.67	19.93	70.19	41.74	41.83

Models	CMU-DoG test set					
	BLEU-2	Rouge-L	Div-2	Average	Extrema	Greedy
ITDD	3.60	-	-	-	-	-
DRD	5.70	-	-	-	-	-
ZRKGC	5.68*	13.05*	8.26*	**66.26***	31.42*	37.91*
KAT	5.81*	11.98*	17.60*	63.72*	35.18*	37.72*
DoHA	6.95*	14.41*	12.14*	65.69	35.51*	39.26*
DIALKI	6.41*	14.64*	**20.43***	63.57*	34.89*	37.60*
CKL	**7.91**	15.87	11.10	65.63	35.81	39.46
w/o $Loss_{KLW}$	7.66	15.75	11.69	65.57*	35.74	**39.48**
w/o $Loss_{CLWR}$	7.62	**15.93**	11.32	65.19	35.71*	39.20*
w/o $Loss_{CLWK}$	7.80	15.91	11.30	65.65	**35.83**	39.47
w/o *CK-Dep*	7.64	15.86	11.67*	65.30*	35.63	39.10*

Table 8.2: Human evaluation on Wizard of Wikipedia test seen and CMU-DoG datasets. The values except for Kappa are in percentage (%).

CKL vs. DIALKI	Relevance				Coherence			
	Win	Loss	Tie	Kappa	Win	Loss	Tie	Kappa
WoW test seen	41.34	25.25	33.41	0.30	42.66	24.26	33.08	0.33
CMU-DoG	49.51	20.79	29.70	0.39	44.55	26.73	28.72	0.45
CKL vs. DIALKI	Informativeness				Overall Preference			
	Win	Loss	Tie	Kappa	Win	Loss	Tie	Kappa
WoW test seen	43.23	24.26	32.51	0.31	36.30	22.28	41.42	0.39
CMU-DoG	48.51	39.61	11.88	0.42	50.50	38.61	10.89	0.41

DIALKI. This indicates that the CKL model improves in terms of relevance, coherence, and informativeness. Fleiss' Kappa (Fleiss and Cohen (1973)) is calculated for each criterion. The resulting Kappa scores are around 0.4, which indicates a moderate agreement among the assessors. From Table 8.2, we can observe that the proposed CKL model outperforms the best baseline DIALKI model from all perspectives.

8.3.2 Ablation Study

Four downgraded versions of CKL are provided. (1) w/o $Loss_{KLW}$, i.e., removing the knowledge latent weight loss function; (2) w/o $Loss_{CLWR}$ (deleting the context latent weight supervision for response generation); (3) w/o $Loss_{CLWK}$ (deleting the context latent weight supervision for knowledge prediction); and (4) w/o *CK-Dep* removes the context-knowledge dependency when generating *KLW*, i.e., removing the Context LWE Cross-Attention module from the *KLW* Generator in Figure 8.3. From Table 8.1, we can observe that all of the BLEU-2 scores decrease. For the WoW dataset, when removing $Loss_{KLW}$ and $Loss_{CLWK}$, the BLEU-2 gets the lowest score among all of the ablation experiments, indicating the importance of knowledge latent weights generation. In terms of the CMU-DoG dataset, which has patterns different from the WoW test seen, w/o $Loss_{CLWR}$ decreases the most. Thus, correctly identifying context seems more crucial than knowledge selection for the CMU-DoG dataset. I presume that CMU-DoG's knowledge sentences are complementary, i.e., different knowledge sentences contain similar information, resulting in the context recognition showing more importance. This assumption is further verified in Sec. 8.3.5. Other metrics decreased to varying degrees but most remain better than the baseline approaches. It is worth noting that while the complete CKL version only exhibits a slight improvement in BLEU-2 scores, it demonstrates a statistical difference in the Average metric, which is closely correlated with human evaluations, as mentioned in Liu et al. (2021e). On the whole, the full version of the CKL performs the best.

8.3.3 Latent Weight Analysis.

To illustrate the effectiveness of the proposed CKL model, I demonstrate: (1) knowledge re-ranking by the knowledge latent weights and (2) Spearman's Correlation between the knowledge latent weights and the pseudo ground truth scores. The WoW test seen set is used for illustration. The same patterns are found for WoW test unseen and CMU-DoG datasets.

WoW and CMU-DoG datasets provide a set of initial knowledge, designated by K. The predicted knowledge latent weights by CKL are scores for each knowledge sentence that can be used to rank knowledge and obtain re-ranked knowledge set K'. Pseudo ground truth knowledge order is constructed by using the response as the query to retrieve from the knowledge named K_{GT}. At this point, the top 1 ranked knowledge sentence in K_{GT} is the most relevant to the response, *Top1Klg*. *P@N* is used as the metric to evaluate the precision. For the samples, the percentage of *Top1Klg* included within the top N-ranked knowledge sentences is computed.

Figure 8.4 shows the results: for the original knowledge order *K*, *P@1* is about 17.5%; for *K'*, *P@1* score is around 30%. For each N, the *P@N* for *K'* is higher than for *K*. That confirms that the latent weight modules can improve the relevance scoring of knowledge sentences.

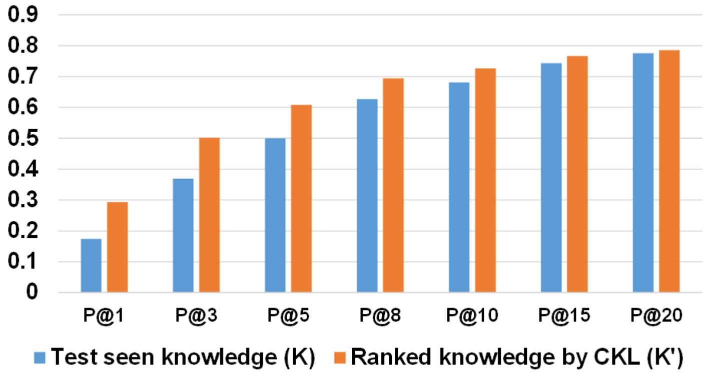


Figure 8.4: *P@N scores for WoW test seen original knowledge set and the re-ranked knowledge set by the proposed CKL.*

Table 8.3: Spearman's Correlation between latent weights and the ground truths. * means significant test value with $p < 0.05$, in comparison with the proposed CKL. 'w/o' mean models without a certain module.

Models	KLW	CLWR	CLWK
ZRKGC	0.1001*	-	-
CKL	0.3700	0.6697	0.6455
w/o $Loss_{KLW}$	0.0966*	0.6699	0.6455
w/o $Loss_{CLWR}$	0.3585	0.4658*	0.6455
w/o $Loss_{CLWK}$	0.3694	0.6696	0.5769*
w/o *CK-Dep*	0.3732	0.6696	0.4816*

Spearman's Correlation

To further analyze the effectiveness of the latent weights *CLWR*, *CLWK* and *KLW*, I calculate Spearman's Correlation between each weight group and the corresponding pseudo ground truths which have been elaborated on in Sec. 8.1 (e.g., for *CLWR* the Spearman's Correlation between *CLWR* and GT_{CLWR} is calculated). The ZRKGC model is used to provide weights for knowledge sentences without context weight (ZRKGC does not provide it) and obtain Spearman's Correlation with the pseudo-ground truth GT_{KLW}. The resulting Spearman's Correlation coefficients with those of the CKL and CKL's ablated models are shown in Table 8.3. For *KLW*, the coefficients are higher than for ZRKGC by a large margin. For CKL's ablated models, the coefficients are lower than for the full CKL model. For instance, *KLW* correlation score 0.0966 for 'w/o $Loss_{KLW}$' is much lower than the score 0.37 for CKL. This further demonstrates all supervised modules are helpful to the entire model.

8.3.4 Low-Resource Experiments

Table 8.4: Wizard of Wikipedia test seen & unseen and CMU-DoG evaluation results on low-resource scenarios. * means significant test value with $p < 0.05$, in comparison with the full version of CKL. All values are expressed as percentages (%).

Models	Wizard of Wikipedia test seen					
	BLEU-2	Rouge-L	Div-2	Average	Extrema	Greedy
Full training data	15.80	23.96	36.36	71.11	42.95	42.54
1/2 training data	13.81*	22.37*	35.79*	70.94	42.30*	42.41
1/4 training data	12.48*	21.17*	34.45*	70.76	42.16*	42.63
1/8 training data	10.38*	18.32*	36.35*	67.84*	40.88*	41.08*
1/16 training data	9.77*	18.84*	31.82*	67.12*	40.03*	40.05*
Zero training data	4.14*	11.44*	25.63*	56.87*	34.90*	35.45*
Models	Wizard of Wikipedia test unseen					
	BLEU-2	Rouge-L	Div-2	Average	Extrema	Greedy
Full training data	16.05	23.93	18.60	70.36	41.93	41.86
1/2 training data	14.09*	22.41*	20.17	69.88*	40.48*	41.75
1/4 training data	12.25*	20.87*	20.93*	69.69*	40.55*	41.70
1/8 training data	10.09*	18.04*	20.58*	65.88*	38.82*	39.92*
1/16 training data	9.64*	18.52*	17.04*	65.56*	38.12*	39.00*
Zero training data	3.96*	11.32*	17.10*	56.36*	33.54*	35.60*
Models	CMU-DoG					
	BLEU-2	Rouge-L	Div-2	Average	Extrema	Greedy
Full training data	7.91	15.87	11.10	65.63	35.81	39.46
1/2 training data	7.25*	14.84*	11.51*	65.07*	34.68*	38.30*
1/4 training data	7.27*	14.20*	10.35*	65.11*	34.90*	39.09*
1/8 training data	6.11*	13.74*	13.00*	62.11*	33.53*	37.16*
1/16 training data	5.68*	13.97*	12.22*	62.05*	34.06*	36.72*
Zero training data	2.51*	8.47*	18.99*	62.65*	30.26*	35.61*

In order to test the CKL's robustness, experiments on low-resource scenarios are examined. Table 8.4 shows the results of the experimental results for low-resource training. We can clearly see the effectiveness of the proposed CKL. For example, the BLEU-2 scores of the CKL model with half of the training data are respectively 13.81%, 14.09%, and 7.25% on the WoW test seen, test unseen, and CMU-DoG datasets, outperforming the best baseline models (DIALKI with 13.72% on WoW test seen set, 13.96% on WoW test unseen set, and DoHA with 6.95% on CMU-DoG). As the scale of the training set decreases, the performance drops gradually. However, when the scale of the training data goes down to less than 1/4 of the original training data, the performance decreases more dramatically. The proposed CKL model performs reasonably well with limited training data, but models with a sufficient amount of training data are still preferred.

8.3.5 Effect of Top-N Retrieved Knowledge

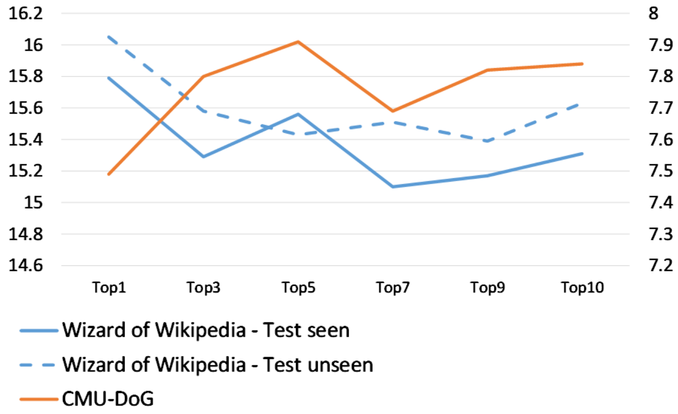


Figure 8.5: *BLEU-2 score (Y-axis) for Wizard of Wikipedia and CMU-DoG under different numbers of top N retrieved knowledge sentences being the ground truth.*

In Sec. 8.1.3, top N retrieved knowledge sentences are set as the ground truth for obtaining GT_{KLW}. How does different N affect will be discussed here. As can be seen in Figure 8.5, a different number of retrieved knowledge sentences, ranging from top 1 to top 10, is investigated. It is clear that for the WoW dataset, using the first retrieved knowledge gets the best results, while for the CMU-DoG dataset, the top-5 group peaks. That indicates that in the WoW dataset, the knowledge other than the top 1 retrieved sentence contains limited useful information. However, the knowledge in the CMU-DoG dataset complements each other. This also explains why when removing $Loss_{CLWR}$, the BLEU-2 decreases most for CMU-DoG (Sec 8.3.2).

8.3.6 Case Study

In order to qualitatively demonstrate the results generated by the CKL and the baseline models, a good case and a bad case generated by different models are provided. The good case is selected by this criterion for a sample: if a knowledge weight latent score is the highest among all of the knowledge sentences and it is also the ground truth knowledge (see latent weight analysis

Table 8.5: A good case and a bad case of the proposed CKL model. In the base case, 'A' and 'B' denote the two Amazon MTurk workers exchanging their opinions.

Good Case
Context: Pop music. I sure am a fan of pop music these days. It is so much fun to listen to! **Ground Truth Response:** Pop music is a genre of popular music that originated in its modern form in the United States, and yes it is fun to listen to.
ZRKGC: I'm a fan of pop music, but it is not the most popular genre. **KAT:** I love pop music too! it originated in the US and UK in the 1950s. **DoHA:** Pop music is a genre of music that originated in the US and Canada. **DIALKI:** Pop and rock are popular music. I like both but I prefer pop music. **CKL:** Pop music is a genre of popular music that originated in its modern form in the United States and the UK during the mid-1950s.
Bad Case
Context: **A:** Swimming is consistently among the top public recreational activities, and in some countries, swimming lessons are a compulsory part of the educational curriculum. **B:** Swimming has been consistently the most popular recreational activity. **A:** It is a fantastic exercise that people can do at almost any age. **Ground Truth Response:** Especially since it mainly relies on the body's natural buoyancy.
ZRKGC: Swimming is consistently the most popular recreational activity. I don't know about it, but in some countries, they have to be a time. **KAT:** It is! Humans can hold their breath underwater and undertake rudimentary locomotive swimming within weeks of birth. **DoHA:** It is! And in some countries, swimming lessons are a compulsory part of the educational curriculum. **DIALKI:** It is amazing that people can do it almost every day, whether it be for recreation, sport, exercise, or survival. **CKL:** I think it is the most popular recreational activity.

in Sec.8.3.3), it is chosen. On the contrary, if the knowledge sentence is the ground truth knowledge but predicted to have the lowest latent score, it is a bad case. From Table 8.5, we can see that in the good case, if the knowledge is predicted correctly, the CKL-generated response is very close to the ground truth target. The other responses produced by the baseline approaches tend to be generic. In terms of the bad case, due to the lack of important information used in the target, the CKL response does not share keywords with the ground truth target even though it is a proper answer for the given context. This comes to the one-to-many problem which describes that many responses are reasonable for a certain dialogue scenario by the natural language. This problem is out of this work's scope.

8.4 Summary

Past studies on capturing context and knowledge relationships to boost the quality of response generation models were restricted to coarse-grain characterization through context-knowledge cross-attention. I recognize the importance of context both for direct contribution to the response generation and for contextualizing the knowledge that can be injected into the response generation process. In this work, I describe the Contextualized Knowledge Learning (CKL) method that incorporates trainable latent vectors and illustrate the CKL by using two vectors which are trained to capture the relationship between context, responses, and the 'best knowledge' (identified through a pre-defined default retrieval process by taking the response as the query) as well as the relationship between contextual knowledge and responses. The trained latent vectors are used to generate latent weights that are used in the loss function and the decoder. With these two mechanisms, the CKL has the flexibility to influence the learning process and has demonstrated superior performance against six strong baselines and reduced datasets. To sum up, the main contributions can be concluded as:

- Differentiated functionality of the context utterances for the knowledge selection task and response generation task, achieved through the technique of training latent vector;
- Latent Weight Enhanced Attention module that incorporates the latent weights into the generation process;
- Effective weak supervision of latent weights training by defining the pseudo ground truths for the context latent weights and knowledge latent weights;
- Robustness of CKL, retaining its effectiveness with reduced amounts of data.

Chapter 9

Conclusions

9.1 Main Conclusions

The focus of this thesis centers on dialogue generation methods. The relevant literature was initially reviewed to establish the foundation for my work within the broader research context. Within this context, five research questions were identified, which, despite numerous notable attempts, have not been fully addressed by the research community. While there have been many commendable efforts in each research direction, significant challenges persist. Solutions to these research questions were explored and empirical findings were presented across Chapters 4 to Chapter 8. In this concluding chapter, a summary of each work will be provided, followed by discussions on potential future research directions and potential workarounds for addressing the identified challenges.

In Chapter 4, my aim is to address RQ 1 (how to affect response generation by incorporating intrinsic dialogue characteristics?) by introducing a context-aware dialogue generation approach called GMATs. The method is inspired by the inherent characteristics of dialogues, such as the role of the speaker and part-of-speech information. Previous research (Clark and Wasow, 1998; Khandelwal et al., 2018; Sankar et al., 2019a) has shown that speakers tend to reuse words they have previously used in the dialogue history and that content words in the conversational context are more important than functional words (as determined by part-of-speech). Building on these findings, the proposed GMATs model includes a speaker-role embedding matrix and a part-of-speech embedding matrix to incorporate dialogue characteristics. Additionally, I investigate four Auxiliary Tasks related to content characteristics (whether a word is a frequent or content word) and speaker characteristics (which speaker the word comes from and whether the word belongs to the post-utterance). Each pair focuses on either general (frequent words and speaker roles) or specific (content words and post-word indicators) aspects. Through ablation studies using two widely-used datasets, DailyDialog and Wizard of Wikipedia, it can be demonstrated that specific aspects have a more substantial impact on BLEU-2 and Meteor scores.

Chapter 5 addresses RQ 2, which pertains to the optimal method for incorporating knowledge and determining the appropriate amount of knowledge to be used. To tackle this challenge, the TED model is introduced, featuring a probability-calculating layer that assigns weights to each injected knowledge unit. The TED model initiates the process by retrieving a set of knowledge sentences using the context as queries. Within the TED decoder, each knowledge sentence's representation is scaled by its weight and subsequently aggregated to yield a unified vector for the final output. Furthermore, an exploration is conducted into the challenge of determining the correct number of knowledge units to employ. Recognizing that the injection of knowledge sentences may introduce both useful information and noise, the hypothesis emerges that not all knowledge sentences should be incorporated. Consequently, a variable number of knowledge sentences is injected, spanning from 1 to 15. This exploration leads to the conclusion that: (1) The suitable number of knowledge units for effective dialogue generation exhibits difference across different datasets. (2) Performance trends follow a similar pattern, characterized by an initial performance increase as the number of injected knowledge units rises, followed by a peak and eventual decline. For instance, in the Wizard of Wikipedia dataset and Reddit dataset, TED achieves its best results with the injection of 3 and

12 knowledge sentences, respectively. This finding raises the significance of knowledge selection as a research question.

Chapter 6 delves into the knowledge selection task (RQ 3), known to significantly enhance response generation when executed effectively (Kim et al., 2020; Lian et al., 2019). To serve dialogue generation, the TPPA model is introduced, functioning as a knowledge-ranking model. The approach unfolds in several steps: Firstly, two distinct knowledge sets are constructed: a post-retrieved knowledge set (obtained by treating the post as the query and using the BM25 algorithm for retrieval) and a response-retrieved knowledge set (similar to the post-retrieved knowledge but with the response as the query). Secondly, an approximation of the prior network to the posterior network is implemented, enabling the post to mimic the ground truth response. This step fosters learning and adaptation within the model. Thirdly, during testing, newly acquired knowledge sentences are scored using the prior network. The retrieval results are compared against robust retrieval-based models, and the selected knowledge is integrated with the generative model, TED. At the time of publication, the outcomes in both retrieval and generation capabilities surpassed existing methods, justifying the TPPA model's effectiveness in knowledge selection.

Chapter 7 addresses RQ 4, which pertains to term-level knowledge de-noising. Previous research has demonstrated that a relevant knowledge set enhances response generation performance, motivating us to investigate fine-grained knowledge de-noising, which involves filtering noise at the term level. The hypothesis is that a term that is semantically closer to the representation of the response is more valuable and should be retained, while others should be discarded. To test this hypothesis, a model called KTWM is proposed, in which a simulated response vector is approximated to the ground truth response's representation. During testing, the simulated response vector replaces the ground truth response to assign weights to each term in the knowledge, thus downgrading noisy terms. Results show that the KTWM effectively produces term weights and improves performance.

Chapter 8 introduces a novel dialogue generation model to address RQ 5. The chapter argues that context plays a crucial role in both knowledge selection and response generation and proposes treating the two tasks separately. To achieve this, the chapter proposes a prompt learning paradigm that generates a context prompt weight for the knowledge and another for the response. This approach is different from previous works that treat the context for those two tasks as the same. The proposed model, CKL, combines sentence-level and term-level weighting in a unified framework, leveraging the attention mechanism to reflect term weights during model training. The results demonstrate that the CKL model significantly outperforms six strong baseline approaches.

In this thesis, five research questions crucial to dialogue generation are explored, and corresponding models are proposed, each tailored to a specific research goal. The objective is to enhance response quality by producing responses with diversity, contextual relevance, and alignment with relevant background knowledge.

9.2 Thinking about Further Works

Multi-turn Context Modeling. As introduced in Chapter 1, five approaches are proposed to tackle five research questions, in which context information plays a critical role, serving as either prompts for response generation or the source for knowledge selection. Typically, this context information consists of several utterances generated by two speakers taking turns. However, most generative models concatenate all context utterances without distinguishing the speakers, resulting in mixed-up responses that lack coherence. For example, in the context of *A: Have you ever seen the Avengers? B: Yes, I went to the cinema to see it when it was released. How do you like it?*, the pronoun “it” from speaker B refers to “Avengers” mentioned by speaker A. While the attention mechanism may capture this relationship, it is not guaranteed and is difficult to explain. The solution to this problem is considered in two stages: (1) modeling the model by speakers. Each speaker has their own utterances, enabling us to obtain a subset of utterances classified by the speaker role; (2) for models with two speakers, explicitly designing attention operations across each term. This approach allows us to model both global information, such as the speaker's talking style, and local information, such as term-level representation updates through attention mechanisms, in the same framework.

Persona and Empathy Injection. The generation of persona-based dialogues involves creating a model that can produce responses with a particular speaking style and reflect human characteristics such as age and hobbies. On the other hand, empathy-based models aim to imbue models with the ability to express emotions. Although these are two distinct tasks, they share a common goal of injecting human-like information and expecting generative models to exhibit human-like characteristics.

The dataset poses a fundamental challenge for both tasks and to address this issue, Zhang et al. (2018b) developed the Persona-Chat dataset. Using Amazon Mechanical Turk (AMT), the authors paired two workers and assigned them a random persona from a pool, resulting in 164,356 utterances in 10,981 dialogues. Persona information in this dataset is expressed through five profile sentences. Another dataset proposed by Rashkin et al. (2019) is the EmpatheticDialogues dataset, consisting of conversations between two individuals, where one expresses a particular emotion and the other responds with empathy. Although many researchers have explored these research directions, several challenges still exist.

- First of all, it lacks diversity in training data. Many of the existing datasets used to train persona-based and empathy-based dialogue models are limited in terms of their diversity. This can result in the models not being able to handle conversations with users from different backgrounds, cultures, and demographics.

- Secondly, the models are unable to handle complex emotions. While some of the existing models are able to generate empathetic responses, they may struggle with more complex emotions that require a deeper understanding of the context and the user's personality.

- When involving persona and empathy information, it is hard for models to maintain consistency, especially if the conversation takes unexpected turns.

Large Language Models (LLMs). Ever since the release of GPT-3 (Brown et al., 2020), the pre-trained models step into the era of large models. GPT-3 boasts an impressive 175 billion parameters, making it the largest pre-trained language model by that period of time. Building upon this milestone, Instruct-GPT (Ouyang et al., 2022) introduced a novel approach called Reinforcement Learning from Human Feedback (RLHF). This involved employing human annotators to provide feedback, effectively using human judgments as the ground truth for training the model through reinforcement learning. This innovative approach yielded the creation of ChatGPT, a widely renowned product [1]. ChatGPT is a versatile, multi-task-oriented tool capable of performing tasks such as information retrieval, code generation, and text generation based on user prompts. Given the remarkable success of ChatGPT, prominent international internet companies, including Google, Meta, and Amazon, have eagerly embraced the trend of developing large language models. This includes the introduction of LaMDA (Thoppilan et al., 2022), PaLM (Chowdhery et al., 2022), and Amazon Titan [2], as they seek to leverage the potential of these expansive language models.

While Large Language Models have significantly propelled research and found applications in our daily lives, there remain several research questions that demand exploration.

[1] https://chat.openai.com/

[2] https://aws.amazon.com/bedrock/titan/

Firstly, training LLMs requires extensive computational resources and time, often requiring thousands of GPUs and several months for model convergence. This high resource requirement poses a barrier to wider participation.

Secondly, ethical concerns are gradually severe. People are increasingly unable to differentiate between text generated by LLMs and human authors, yet we cannot guarantee the accuracy of LLM-generated outputs. This raises the risk of knowledge pollution on the internet, as LLMs are trained on large-scale datasets extracted from the web, potentially perpetuating misinformation. Addressing these ethical issues is both urgent and imperative.

Lastly, there remains a lack of robust evaluation methods for LLMs. Current prevalent metrics for assessing dialogue models rely on lexical-based metrics (e.g., BLEU, Meteor) and embedding-based metrics (such as Average, Bert-Score), all of which require references for calculation. However, in real-life dialogue scenarios, one query can have multiple valid responses, making evaluation challenging. Employing human annotators to assess all LLM-generated outputs is impractical. Furthermore, if evaluation methods are ambiguous, the correctness of LLM-generated outputs becomes uncertain, exacerbating the ethical dilemma.

To address these limitations, future research could focus on the following areas:

(1) Developing more diverse datasets. A potential task is to work on creating more diverse datasets that include conversations with users from different backgrounds, cultures, and demographics. This could help improve the generalizability of persona-based and empathy-based dialogue models.

(2) Incorporating reinforcement learning techniques. Reinforcement learning techniques could be used to improve the consistency of persona-based and empathy-based dialogue models. By providing rewards for consistent responses, the models could learn to maintain the persona of the user throughout the conversation.

(3) Integrating multi-modal information. To better understand user emo-

tions and needs, future works could explore the integration of multi-modal information, such as facial expressions and voice intonations, into persona-based and empathy-based dialogue models.

(4) Investigating the interpretability of LLMs, it remains unclear why LLMs generate responses in their current manner. When provided with prompts, these prompts are converted into vectors and undergo internal calculations within the LLMs. The resulting outputs are determined based on the highest probabilities associated with each generated token. The reasons behind the selection of specific tokens remain a relatively unexplored area of study. Some potential research questions in this domain include, but are not limited to: a) How do LLMs store and access knowledge? b) How do LLMs process prompts and extract the relevant information for generating responses? c) What parameter-related aspects enable large models to outperform smaller ones?

(5) Exploring evaluation techniques for LLMs presents a challenge due to the "one-to-many" phenomenon, making traditional assessment methods inappropriate. Similar to the way humans learn, LLMs require guidance and supervision in their training and behavior. One potential approach involves training teacher LLMs that are provided with 100% accurate information, which can then be used to supervise and instruct student LLMs.

In summary, while the existing works on persona-based and empathy-based dialogue generation are promising, there is still significant room for improvement. To gain deeper insights into Large Language Models and exert control over them, we need to focus on addressing challenges such as the explanation and evaluation issues. By addressing some of the limitations outlined above, future works could develop more advanced models that are better able to handle complex conversations with users from diverse backgrounds.

In the end, I hope this thesis could provide inspiration to the researchers in the dialogue generation community.

Bibliography

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. Cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. Plato: Pretrained dialogue generation model with discrete latent variable. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 85–96.

Antoine Bordes, Y.-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *ICLR*. OpenReview.net.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111.

Kyunghyun Cho, Bart van Merriënboer, Çağlar Gu'lçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. 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.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Herbert H Clark and Thomas Wasow. 1998. Repeating words in spontaneous speech. *Cognitive psychology*, 37(3):201–242.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Mostafa Dehghani, Hosein Azarbonyad, Jaap Kamps, and Maarten de Rijke. 2019a. Learning to transform, combine, and reason in open-domain question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 681–689. ACM.

Mostafa Dehghani, Hosein Azarbonyad, Jaap Kamps, and Maarten de Rijke. 2019b. Learning to transform, combine, and reason in open-domain question answering. In *WSDM*, pages 681–689. ACM.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2019c. Universal transformers. In *International Conference on Learning Representations*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations (ICLR)*.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.

Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. Bootstrapping dialog systems with word embeddings.

Michel Galley, Chris Brockett, Xiang Gao, Jianfeng Gao, and Bill Dolan. 2019. Grounded response generation task at dstc7. In *AAAI Dialog System Technology Challenges Workshop*.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715.

Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. 2016. Professor forcing: a new algorithm for training recurrent networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4608–4616.

A. Graves. 2016. Adaptive Computation Time for Recurrent Neural Networks. *arXiv e-prints*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. 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)*, pages 1631–1640.

Xiaodong Gu, Kang Min Yoo, and Sang-Woo Lee. 2021. Response generation with context-aware prompt learning. *arXiv preprint arXiv:2111.02643*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338. ACM.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. *stat*, 1050:5.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proc. of the 56th Ann. Meeting of the Assoc. for Comp. Ling. (Volume 1: Long Papers)*, pages 284–294.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential Latent Knowledge Selection for Knowledge-Grounded Dialogue. In *ICLR*.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Hailiang Li, YC Adele, Yang Liu, Du Tang, Zhibin Lei, and Wenye Li. 2019a. An augmented transformer architecture for natural language generation tasks. In *2019 Int. Conf. on Data Mining Workshops (ICDMW)*, pages 1–7. IEEE.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017a. Learning through dialogue interactions by asking questions. In *ICLR*. OpenReview.net.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.

Linxiao Li, Can Xu, Wei Wu, Yufan Zhao, Xueliang Zhao, and Chongyang Tao. 2020. Zero-resource knowledge-grounded dialogue generation. *Advances in Neural Information Processing Systems*, 33:8475–8485.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017b. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proc. of the Eighth Int. Joint Conf. on Natural Language Proc. (Volume 1: Long Papers)*, pages 986–995.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019b. Incremental transformer with deliberation decoder for document grounded conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21.

Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to select knowledge for response generation in dialog systems. In *IJCAI International Joint Conference on Artificial Intelligence*, page 5081.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yanxiang Ling, Fei Cai, Xuejun Hu, Jun Liu, Wanyu Chen, and Honghui Chen. 2021. Context-controlled topic-aware neural response generation for open-domain dialog systems. *Info. Processing & Management*, 58(1):102392.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016a. 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.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016b. 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.

Longxiang Liu, Zhuosheng Zhang, Hai Zhao, Xi Zhou, and Xiang Zhou. 2021a. Filling the gap of utterance-aware and speaker-aware representation for multi-turn dialogue. In *The Thirty-Fifth AAAI Conf. on Artificial Intelligence (AAAI-21)*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Shilei Liu, Xiaofeng Zhao, Bochao Li, Feiliang Ren, Longhui Zhang, and Shujuan Yin. 2021c. A three-stage learning framework for low-resource knowledge-grounded dialogue generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2262–2272.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021d. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zeyang Liu, Ke Zhou, and Max L Wilson. 2021e. Meta-evaluation of conversational search evaluation metrics. *ACM Transactions on Information Systems (TOIS)*, 39(4):1–42.

Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao, and Michel Galley. 2017. Multi-task learning for speaker-role adaptation in neural conversation models. In *IJCNLP(1)*, pages 605–614. Asian Federation of Natural Language Processing.

Minh-Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. 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 11–19.

Fenglong Ma, Radha Chitta, Saurabh Kataria, Jing Zhou, Palghat Ramesh, Tong Sun, and Jing Gao. 2017. Long-term memory networks for question answering. *CoRR*, abs/1707.01961.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018a. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478, Melbourne, Australia. Association for Computational Linguistics.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018b. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478, Melbourne, Australia. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *proceedings of ACL-08: HLT*, pages 236–244.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332.

Jan Niehues and Eunah Cho. 2017. Exploiting linguistic resources for neural machine translation using multi-task learning. In *Proc. of the Second Conf. on Machine Translation*, pages 80–89.

Hiroki Ouchi and Yuta Tsuboi. 2016. Addressee and response selection for multi-party conversation. In *Proc. of the 2016 Conf. on Empirical Methods in Natural Language Processing*, pages 2133–2143.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ashwin Paranjape, Omar Khattab, Christopher Potts, Matei Zaharia, and Christopher D Manning. 2021. Hindsight: Posterior-guided training of retrievers for improved open-ended generation. *arXiv preprint arXiv:2110.07752*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Shrimai Prabhumoye, Kazuma Hashimoto, Yingbo Zhou, Alan W Black, and Ruslan Salakhutdinov. 2021. Focused attention improves document-grounded generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4274–4287.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Hannah Rashkin, David Reitter, Gaurav Singh Tomar, and Dipanjan Das. 2021. Increasing faithfulness in knowledge-grounded dialogue with controllable features. In *Proc. of the 59th Annual Meeting of the Assoc. for Comp. Ling. and the 11th Int. Joint Conf. on Natural Language Proc. (Volume 1: Long Papers)*, pages 704–718.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR‘‘94*, pages 232–241. Springer.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Vasile Rus and Mihai Lintean. 2012. An optimal assessment of natural language student input using word-to-word similarity metrics. In *Intelligent Tutoring Systems: 11th International Conference, ITS 2012, Chania, Crete, Greece, June 14-18, 2012. Proceedings 11*, pages 675–676. Springer.

Chinnadhurai Sankar, Sandeep Subramanian, Christopher Pal, Sarath Chandar, and Yoshua Bengio. 2019a. Do neural dialog systems use the conversation history effectively? an empirical study. In *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 32–37.

Chinnadhurai Sankar, Sandeep Subramanian, Christopher Pal, Sarath Chandar, and Yoshua Bengio. 2019b. Do neural dialog systems use the conversation history effectively? an empirical study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 32–37.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proc. of the First Conf. on Mach. Transl.: Vol 1*, pages 83–91.

Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proc. of the AAAI Conf. on Artificial Intelligence*, volume 30.

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

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *EMNLP*, pages 2210–2219. Association for Computational Linguistics.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 101–110. ACM.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Shijing Si, Dinghan Shen, Dong Wang, and Lawrence Carin. 2021. Syntactic knowledge-infused transformer and bert models. In *CIKM Workshops*.

Yik-Cheung Tam. 2020a. Cluster-based beam search for pointer-generator chatbot grounded by knowledge. *Computer Speech & Language*, 64:101094.

Yik-Cheung Tam. 2020b. Cluster-based beam search for pointer-generator chatbot grounded by knowledge. *Computer Speech & Language*, 64:101094.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Zhiliang Tian, Rui Yan, Lili Mou, Yiping Song, Yansong Feng, and Dongyan Zhao. 2017. How to make context more useful? an empirical study on context-aware neural conversational models. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–236.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Anran Wang, Anh Tuan Luu, Chuan-Sheng Foo, Hongyuan Zhu, Yi Tay, and Vijay Chandrasekhar. 2018. Holistic multi-modal memory network for movie question answering. *CoRR*, abs/1811.04595.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *3rd International Conference on Learning Representations, ICLR 2015.*

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2021a. Alternating recurrent dialog model with large-scale pre-trained language models. In *Proc. of 16th Conf. of the European Chapter of the Assoc. for Comp. Ling.: Main Volume*, pages 1292–1301.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ‘ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Zeqiu Wu, Bo-Ru Lu, Hannaneh Hajishirzi, and Mari Ostendorf. 2021b. Dialki: Knowledge identification in conversational systems through dialogue-document contextualization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1852–1863.

Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. 2018. Hierarchical recurrent attention network for response generation. In *Proc. of the AAAI Conf. on Artificial Intelligence*, volume 32.

Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology.*

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *Proc. of 22nd Int. Conf. on World Wide Web*, pages 1445–1456.

Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *SIGIR*, pages 245–254. ACM.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Semih Yavuz, Abhinav Rastogi, Guan-Lin Chao, and Dilek Hakkani-Tur. 2019. Deepcopy: Grounded response generation with hierarchical pointer networks. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 122–132.

Hao-Tong Ye, Kai-Lin Lo, Shang-Yu Su, and Yun-Nung Chen. 2020. Knowledge-grounded response generation with deep attentional latent-variable model. *Computer Speech & Language*, 63:101069.

Koichiro Yoshino, Chiori Hori, Julien Perez, Luis Fernando D''Haro, Lazaros Polymenakos, Chulaka Gunasekara, Walter S Lasecki, Jonathan K Kummerfeld, Michel Galley, Chris Brockett, et al. Dialog system technology challenge 7.

Hainan Zhang, Yanyan Lan, Liang Pang, Hongshen Chen, Zhuoye Ding, and Dawei Yin. 2020. Modeling topical relevance for multi-turn dialogue generation. In *Proc. of the 29th Int. Joint Conf. on Art. Intell., IJCAI-20*, pages 3737–3743. Int. Joint Conf.s on Artificial Intelligence Organization.

Hainan Zhang, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. 2019a. Recosa: Detecting the relevant contexts with self-attention for multi-turn dialogue generation. In *Proc. of the 57th Annual Meeting of the Assoc. for Comp. Ling.*, pages 3721–3730.

Rui Zhang, Honglak Lee, Lazaros Polymenakos, and Dragomir Radev. 2018a. Addressee and response selection in multi-party conversations with speaker interaction rnns. In *Thirty-Second AAAI Conf. on Artificial Intelligence*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018b. 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. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Xueliang Zhao, Wei Wu, Chongyang Tao, Can Xu, Dongyan Zhao, and Rui Yan. 2020a. Low-resource knowledge-grounded dialogue generation. In *International Conference on Learning Representations (ICLR)*.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020b. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390.

Yufan Zhao, Can Xu, and Wei Wu. 2020c. Learning a simple and effective model for multi-turn response generation with auxiliary tasks. In *Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3472–3483.

Chujie Zheng and Minlie Huang. 2021. Exploring prompt-based few-shot learning for grounded dialog generation. *arXiv preprint arXiv:2109.06513*.

Wen Zheng, Natasa Milic-Frayling, and Ke Zhou. 2020. Approximation of response knowledge retrieval in knowledge-grounded dialogue generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3581–3591.

Wen Zheng, Nataša Milić-Frayling, and Ke Zhou. 2021. Knowledge-grounded dialogue generation with term-level de-noising. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2972–2983.

Wen Zheng and Ke Zhou. 2019. Enhancing conversational dialogue models with grounded knowledge. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 709–718.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In *Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing*, pages 708–713.

C. Zhu, M. Zeng, and X. Huang. 2018. SDNet: Contextualized Attention-based Deep Network for Conversational Question Answering. *arXiv e-prints*.

Appendix A

List of Abbreviations

RQ	Research Question
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long-Short Term Memory
GRU	Gated Recurrent Units
RL	Reinforcement Learning
MemNN	Memory Neural Network
TF-IDF	Term Frequency - Inverse Document Frequency
E2E	End to End
MLP	Multi-Layer Perceptrons
MSE	Mean Squared Error
BCE	Binary Cross Entropy
PoS	Part-of-Speech
Seq2Seq	Sequence to Sequence model
SRV	Simulated Response Vectors
PRK	Post-Retrieved Knowledge
RRK	Response-Retrieved Knowledge
DSTC	Dialog System Technology Challenges
GT	Ground Truth
UWP	Useful Words Proportion
KPW	Knowledge Prompt Weight
CPWR	Context Prompt Weight for Response
CPWK	Context Prompt Weight for Knowledge
BLEU	Bilingual Evaluation Understudy
METEOR	Metric for Evaluation of Translation with Explicit ORdering
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
BS	Bert Score
Div-1	Uni-gram Distinct score
Div-2	Bi-gram Distinct score
WoW	Wizard of Wikipedia dataset
BERT	Bidirectional Encoder Representations from Transformers
GMATs	Generative Model with Auxiliary Tasks
TED	Transformer with Expanded Decoder
TPPA	Transformer & Post based Posterior Approximation
KTWM	Knowledge Term Weighting Model
CKL	Contextual Knowledge Learning