

**TOWARDS INTELLIGENT CONVERSATIONAL ASSISTANTS: ENHANCING
TASK-ORIENTED DIALOGUE SYSTEMS WITH KNOWLEDGE
INTEGRATION**

A Dissertation
Presented to
The Academic Faculty

By

Ruolin Su

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical & Computer Engineering
Department of Electrical & Computer Engineering

Georgia Institute of Technology

May 2025

© Ruolin Su 2025

**TOWARDS INTELLIGENT CONVERSATIONAL ASSISTANTS: ENHANCING
TASK-ORIENTED DIALOGUE SYSTEMS WITH KNOWLEDGE
INTEGRATION**

Thesis committee:

Dr. Biing-Hwang Juang
Electrical & Computer Engineering
Georgia Institute of Technology

Dr. Mark Davenport
Electrical & Computer Engineering
Georgia Institute of Technology

Dr. David Anderson
Electrical & Computer Engineering
Georgia Institute of Technology

Dr. Yao Xie
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Chin-Hui Lee
Electrical & Computer Engineering
Georgia Institute of Technology

Date approved: April 14, 2025

Real knowledge is to know the extent of one's ignorance.

知之為知之，不知為不知，是知也。

Confucius

ACKNOWLEDGMENTS

The completion of my PhD journey has been an extraordinary and transformative experience. It would not have been possible without the support, guidance, and encouragement of many important people in my life. I would like to take this opportunity to express my deepest gratitude to my advisor, Prof. Biing-Hwang Juang. Thank you for providing me with the space to explore my ideas, the encouragement to push my boundaries, and the wisdom to navigate challenges with resilience. Your belief in my potential has been a source of constant motivation, and I am truly grateful for the opportunity to learn under your mentorship.

I would also like to express my gratitude to my dissertation committee members, Prof. David Anderson, Prof. Chin-Hui Lee, Prof. Mark Davenport, and Prof. Yao Xie for their time, expertise, and valuable feedback. I deeply appreciate your willingness to engage with my research. I am thankful to my colleagues and collaborators, Ting-Wei Wu, Jingfeng Yang, Zhongkai Sun, Cynthia Lu, Chengyuan Ma, Yunji Wang, Linfeng Song, John Chen, Hao Tian, Yu Zhang, Yiqi Feng, Andrew Ding; and people who have helped me on this journey, Prof. Diyi Yang, Prof. Xiangwei Zhou, Prof. Nilanjan Sarkar, Prof. Xinyu Chen, Prof. Xuedong Huang, Min-Hung Chen, Ran Xu, Yinghong Lan, Jinyu Li, Zhong Meng, Chao Weng, Hao Ye, Jinfeng Rao, Liangbei Xu, Yuxin Wu, Zhehui Chen. Special thanks to my manager, Anjan Goswami, for his generous support to the final completion of this thesis.

A heartfelt thank you to my family, whose unwavering support and encouragement have been my foundation. To my parents, Xianjun Su and Wenjie Hu, your belief in me and your sacrifices have made this achievement possible. To my dear friends, Mingyi Liu, Lijing Zhai, Yifu Sun, Xueyao Zhao, Nana Wang, Yuchen Yang, Dichen Liu, Ningwei Jiang, Pengfei Chen, Chenhui Qian, Xinzhu Bei, Zhengzhe Yang, Chen Feng, Haoming Jiang, Yepu Cui, Qi Zhou, Wenqi Wei, Yi Zhou, Xueyang Xu, Mingdi Mao, Wei Deng,

Xinyi Wang, Guoyuan Wu, Mingyu Guan, Bing He, Ke Xu, Qianyun Chen, Yunhai Han, Dayu Zhu, Shuang Yao, Haoze Wu, your support and understanding have been essential in maintaining balance throughout this journey. I am thankful to my boyfriend, Chengqi Huang, for endless encouragement and for always cheering me on through every milestone.

I am profoundly grateful to each and every one of you who has contributed to this journey in one way or another. Thank you.

TABLE OF CONTENTS

Acknowledgments	iv
List of Tables	xi
List of Figures	xiv
List of Acronyms	xvi
Summaryxviii
Chapter 1: Introduction	1
1.1 Overview	1
1.2 Motivation of Specialized Knowledge in TOD Systems	5
1.2.1 An Analogy of Human Expertise and Intelligent Assistants	5
1.2.2 The Revolution of LLMs on TOD Research	6
1.2.3 Why Task-Oriented Dialogue Systems Still Matter in the Age of LLMs?	6
1.3 Thesis Outline	7
Chapter 2: Preliminaries in Discourse Theory and Computational Dialogue Systems	10
2.1 Foundations of Computational Discourse Analysis	10
2.1.1 Taxonomy of Communication Systems	11

2.1.2	Variability of Dialogue Intuitive	12
2.1.3	Dialogue Acts	13
2.2	Computational Dialogue Systems	14
2.2.1	Overview of Dialogue Systems	14
2.2.2	Dialogue States	15
2.2.3	Finite state-based systems	18
2.2.4	Form-based systems	19
2.2.5	Agent-based systems	19
2.2.6	Challenges in Task-Oriented Dialogue Systems	19
2.3	Knowledge in TOD Systems	20
2.3.1	Domain-Specific Knowledge	20
2.3.2	Dialogue-Level Knowledge	21
2.3.3	Cross-lingual Knowledge	21
2.4	Using Large Language Models in Dialogue Systems	22
2.4.1	Applications of Large-Language Models	22
2.4.2	Understanding Complex Data Structures Using LLMs	22
2.4.3	Domain Adaptation Using LLMs	23
Chapter 3: Methodology for Knowledge Integration in Task-Oriented Dialogue Systems		24
3.1	Proposed Methodologies	24
3.1.1	Domain-Specific Knowledge	24
3.1.2	Dialogue-Level Knowledge	26
3.1.3	Cross-Lingual Knowledge	27

3.1.4	Applications	27
3.2	Conclusion	29
Chapter 4: Incorporating Domain-Specific Knowledge for Task-Oriented Dialogue System		31
4.1	Introduction	31
4.2	Dialogue State Tracking with Slot-Value Transfer	32
4.2.1	Motivation	32
4.2.2	Method	34
4.2.3	Experiments	36
4.2.4	Results	37
4.2.5	Discussion	38
4.3	Dialogue State Tracking Based on Schema-Guided Knowledge Graphs	39
4.3.1	Motivation and Contribution	39
4.3.2	Method	42
4.3.3	Experiment	46
4.3.4	Results and Analyses	49
4.4	Conclusion	54
Chapter 5: Incorporating Dialogue-Level Knowledge for Task-Oriented Dialogue System		56
5.1	Introduction	56
5.2	Incorporating Knowledge from Dialogue Acts	57
5.2.1	Motivation	57
5.2.2	Model	60

5.2.3	Results and Analysis	62
5.3	Incorporating Knowledge from Task-Oriented Dialogue Pipeline	64
5.3.1	Preliminaries	66
5.3.2	Method	68
5.3.3	Experiment	70
5.3.4	Results & Discussion	73
5.3.5	Time Complexity Analysis	77
5.3.6	Impact of Expert Numbers	77
5.4	Conclusion	78
Chapter 6: Enhancing Commonsense Reasoning Across Languages		80
6.1	Introduction	80
6.2	Attention-Based Cross-Lingual Commonsense Knowledge Transfer	83
6.2.1	Problem Definition	83
6.2.2	Step One: Task-Adaptive Pre-Training	84
6.2.3	Step Two: Commonsense Differentiation	84
6.2.4	Step Three: Knowledge-Transfer Fine-Tuning	86
6.3	Experiments and Analyses	86
6.3.1	Experimental Results	88
6.3.2	Discussion	89
6.3.3	Attention Heatmap from X-CODAH	90
6.4	Conclusion	91
6.5	Conclusion	92

Chapter 7: Conclusion	94
7.1 Summary of Contributions	94
7.2 Limitations of Study	96
7.3 Discussions and Future Work	98
7.3.1 Transfer learning for large-scale task-oriented DST	98
7.3.2 Structured DST with domain reasoning	98
7.3.3 Proposing a task for task-oriented dialogue simulation	99
Appendices	100
Appendix A: Experimental Details	101
Appendix B: Datasets	104
Appendix C: Analyses	105
References	107
Vita	127

LIST OF TABLES

4.1	Zero-shot joint goal accuracy on MultiWoz 2.1 [159]. Model results of TRADE [58], MA-DST [162], and SUMBT [161] are obtained from reference papers. TransferQA (T5-Small) follows all setups in [151], but is trained using T5-Small.	37
4.2	Ablation study on the two components of the choice-fusion mechanism: appreciative choice selection with KLD loss (KLD) and context-choice fusion (Fuse). T5-Small [155] is used and the evaluation results on joint goal accuracy (JGA), slot goal accuracy (SGA), and F1 in five domains of MultiWoz 2.1.	38
4.3	Accuracy on each domain on the SGD test set. AdapterCL [†] is drawn from their best model’s results, and Prompt-Tuning* is drawn from our re-implementation.	46
4.4	Accuracy of our model and baselines on SGD dataset by domains. Our model outperforms baselines on joint and slot goal accuracies. AdapterCL [†] is drawn from their best model’s results, and Prompt-Tuning* is drawn from our re-implementation.	47
4.5	Joint and slot goal accuracy on each domain on MultiWOZ2.1.	49
4.6	Ablation results on the first 15 services of SGD listed in Table B.3, section B.2. All numbers are reported in average joint goal accuracy(JGA) (%) over 15 tasks. <i>w/o Active w/o Active</i> means the graph module encodes the same schema information in training. <i>w/o SlotConnect</i> indicates the model is trained without graph modules while only adding the same amount of prompt tokens as graph modules.	52
5.1	An example of cross-domain dialogue with dialogue state representations and system dialogue acts in MultiWOZ 2.1	57
5.2	System Dialogue Acts in MultiWOZ 2.1, consisting of 13 acts serving illocutionary functions in dialogues for reservation service across 7 domains . .	59

5.3	Joint and Slot Goal Accuracy on MultiWOZ 2.1	63
5.4	Ablation Study on MultiWOZ 2.1 Dev Set	64
5.5	Summary of MultiWOZ2.1 dataset [159]. <i>#non-cat</i> and <i>#cat</i> are the total numbers of non-categorical and categorical slots. Avg. Turns are the average number of turns per dialogue, and Avg. Active are the average number of active (NOT none) slots of each turn.	67
5.6	Representations of input and output in task-oriented dialogue system.	69
5.7	Accuracy (%) on three intent prediction datasets with full-data experiments. \diamond comes from Casanueva <i>et al.</i> (2020). All others are reported as in the original papers. Models with * are classification-based.	71
5.8	Joint Goal Accuracy (%) for DST on MultiWOZ 2.1 and 2.2. Results with \ddagger are from He <i>et al.</i> (2022). \dagger represents the results of our re-implementation. All others are reported as in the original papers.	72
5.9	Evaluation of NLG on Inform, Success, BLEU, and Combined Scores, where Combined = (Inform + Success) \times 0.5 + BLEU.	73
5.10	Comparison of the inference time with small and base-size models of PP-TOD and TOATOD for NLG on MultiWOZ 2.1. All models are experimented with 5 same and randomly sampled batch sizes. Average time is reported. \downarrow : Smaller is better.	77
6.1	Accuracy on the test set of X-CSQA and X-CODAH in German. MCP(XLM-R _B)* model is trained in English and machine-translated German. The <i>align</i> , <i>diff</i> , <i>nc</i> refer to the objectives in equation (Equation 6.4), (Equation 6.5), and (Equation 6.6), respectively.	88
6.2	Dev accuracy on X-CSQA taking commonsense or non-commonsense embeddings as inputs for the classifier.	91
A.1	Statistics of the domain-agnostic question-answering (QA) dataset.	101
A.2	The slot goal accuracy of non-categorical and categorical slots on MultiWoz 2.1, respectively. <i>#non-cat</i> and <i>#cat</i> are the total numbers of non-categorical and categorical slots for dev.	102
A.3	Typical errors generated by the model on MultiWoz 2.1 dev set.	103

B.1	Statistics of SGD and MWOZ 2.1.	104
B.2	Statistics of datasets for CLICKER.	104
B.3	Service names and corresponding domains in Schema-Guided-Dialogue. . .	104
C.1	Mean and standard deviation of all reported scores in Table Table 5.8 and Table Table 5.9 using 5 randomly sampled batch sizes, which are the same for all models and datasets. Student paired t-test shows $p < 0.01$ for scores changed by batch size. Combined = (Inform + Success) \times 0.5 + BLEU. . .	105
C.2	Ablation on the positions of SoftMoE layer in T5.	106

LIST OF FIGURES

1.1	Pipeline of a task-oriented dialogue system	3
1.2	Illustration of the proposed knowledge integration approaches in this thesis.	7
2.1	An example of a task-oriented dialogue with slot-value representations . . .	16
3.1	Methodology for intervening dialogue state tracking at different levels. . . .	25
4.1	The overview of CoFunDST. The input is a concatenated query consisting of a question (Tangerine) and reference context (Blue). A sequence of choice tokens (Orange) is used for training and reference. The golden answer token (Yellow) is an extra input during training as extra supervision for encoding the concatenated query.	33
4.2	Illustration of prompt-based adaptation methods: natural language prompt, in-context learning, and schema graph-induced prompt.	40
4.3	The overview of schema graph-induced prompt (SHEGO) architecture for multi-domain DST.	43
4.4	Comparison of architectures of graph neural networks. Results are shown in Average JGA over the first 15 services in SGD as listed in Table B.3. . .	51
5.1	Act-Aware Model Architecture for Dialogue State Tracking.	61
5.2	Visualization of attention weights on the dialogue act attention layer of our model for categorical slots.	65

5.3	Architecture of the SMETOD as in Transformer encoders. The result from the DB derived from the output of DST is used for NLG inference. All of the expert layers share the same architecture. The input is ensembled by experts in the Soft-MoE layer for improving model capacity without the cost of efficiency. The model is fine-tuned by maximizing the likelihood of predicting the next token for NLU, DST, and NLG outputs.	66
5.4	(Top) Performance of SMETOD as a function of the number of experts, for models with a fixed number of experts \times slots-per-expert. (Bottom) Performance of SMETOD trained with increased experts and 2 slots per expert. JGA and F1 scores are on MultiWOZ 2.1 dev set for DST.	76
6.1	The overview of CLICKER framework, which is trained in three steps with joint objectives for cross-lingual commonsense reasoning. Parameters in XLM-R and self-attention layers are shared in all steps. Each ✓ or ✗ represents whether the following choice is commonsense reasonable or not, respectively.	81
6.2	Attention head of the self-attention layer. The given example is from X-CSQA.	90
6.3	Attention head of the self-attention layer. The given example is from X-CODAH.	91

LIST OF ACRONYMS

ADST	Act-aware Dialogue State Tracker
CIP	Conversational Input Processor
CLICKER	Cross-LIngual Commonsense Knowledge transfER
CLK	Cross-Lingual Knowledge
CoFunDST	Choice-Fusion Dialogue State Tracking
CSR	Commonsense Reasoning
DLK	Dialogue-Level Knowledge
DM	Dialogue Management
DSK	Domain-Specific Knowledge
DST	Dialogue State Tracking
IA	Intelligent Assistant
LLM	Large Language Model
MoE	Mixture-of-Experts
MRC	machine reading comprehension
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
NMT	Neural Machine Translation
PLM	Pre-trained Language Model
QA	Question Answering
RST	Rhetorical Structure Theory
SGD	Schema-Guided Dialogue dataset

SHEGO ScHEma Graph-guided prOmpt

SMETOD Soft Mixture-of-Expert Task-Oriented Dialogue

SOTA state-of-the-art

TOD Task-Oriented Dialogue

SUMMARY

This thesis explores the integration of knowledge into Task-Oriented Dialogue (TOD) systems within Intelligent Assistants (IA), enhancing their ability to understand user intent and execute structured tasks. While dialogue systems have evolved significantly with advancements in Natural Language Processing (NLP) and Large Language Models (LLMs), TOD systems still face challenges in knowledge integration, scalability, and generalization. This work proposes methodologies for incorporating domain-specific, dialogue-level, and cross-lingual knowledge into TOD systems to improve their adaptability and effectiveness across diverse applications.

The study identifies key challenges in knowledge integration, emphasizing the need for specialized domain knowledge to improve Dialogue State Tracking (DST) and response generation. To address these challenges, this thesis introduces innovative methods such as structured slot-value transfer and schema-guided knowledge graphs, enhancing both accuracy and scalability in dialogue management. Specifically, we propose choice-fusion DST and schema graph-guided prompts for enhance state tracking and facilitate domain adaptation. Furthermore, it explores the integration of dialogue-level knowledge to improve context awareness, such as incorporating dialogue acts into slot-value prediction for enhanced comprehension and employing a soft mixture-of-experts approach to develop more efficient and scalable TOD systems.

Additionally, the thesis presents a cross-lingual knowledge transfer mechanism to improve commonsense reasoning in low-resource languages, enhancing the multilingual adaptability of TOD systems. These contributions collectively advance the design of intelligent conversational agents by enabling systematic knowledge integration and improving the efficiency of dialogue systems in real-world applications.

Overall, this research establishes a comprehensive framework for knowledge-enhanced TOD systems, addressing critical limitations in current approaches. By integrating struc-

tured knowledge across different levels, it enhances system performance in user intent recognition, dialogue state tracking, and response generation. The findings contribute to the broader field of conversational AI, providing scalable and adaptable solutions for intelligent assistants in diverse domains and multilingual settings. They hold great promise for refining knowledge integration techniques, expanding cross-lingual capabilities, and exploring further applications of LLMs in TOD systems.

In summary, this work focuses on methodologies for integrating domain-specific, dialogue-level, and cross-lingual knowledge into dialogue systems, evaluating its impact across diverse contexts to enhance the design of more effective and intelligent conversational agents. By equipping TOD systems with mechanisms for dynamic knowledge incorporation, it fosters the development of more robust, scalable, and adaptable conversational agents. These advances contribute to improving user interaction, improving response accuracy, and broadening the applicability of TOD systems across various domains and languages.

CHAPTER 1

INTRODUCTION

1.1 Overview

The growing demand for computers to better "understand" their users, enabling customization and prioritization of information and communication, has driven the rise of **Intelligent Assistant (IA)** [1]. These systems mitigate information and communication overload by organizing and prioritizing data based on user needs. Since the early 21st century, they have evolved alongside advancements in call filtering and web search services, enhancing natural intelligence through context-aware processing and adaptive information management. Of considerable interest is the ability to develop techniques to extract important information from interactions with humans, to timely understand or infer the situation, and to decide on an appropriate responsive action to complete specific tasks. An IA enhances this capability by combining human-like reasoning with abstract information and providing tools for efficiently building complex applications, such as manufacturing or service processes.

Natural language dialogue in spoken or written form is the primary mode of human-machine communication in IA systems[2]. Recent advancements in prompt engineering, particularly with models like ChatGPT¹ have enhanced the effectiveness of dialogue systems by optimizing input prompts to guide model responses. Dialogue is characterized by its structured information exchange, including implicit rules for turn-taking, and typically limited to two participants. Moreover, it involves both domain-specific and general knowledge - features that present significant technological challenges and remain underexplored. Additionally, dialogue is goal-oriented, providing a clear basis for evaluating the

¹<https://chat.openai.com/chat>

performance of intelligent systems. In contrast, conversation is an informal exchange of thoughts, ideas, opinions, emotions, or information through spoken words between two or more participants, often without a predefined objective.

Accordingly, computational systems that support conversational services can be categorized into two distinct diagrams based on their application scenarios. Closed-domain dialogue systems are designed for specific task-oriented goals, facilitating focused and purpose-driven interactions. These systems are widely used in applications such as ticket booking, online shopping, travel planning, and recommendation services. Additionally, dialogue systems operate as sequential decision-making models, requiring the ability to anticipate future interactions and plan responses accordingly [3]. The class of closed-domain dialogue systems designed to assist humans in accomplishing predefined tasks through machine interaction is known as **Task-Oriented Dialogue (TOD) systems** or **transactional dialogue systems**. On the other hand, open-domain conversational systems, commonly referred to as **chatbots** or **chit-chat agents**, engage in unrestricted, free-flowing conversations without a predefined objective. These systems allow conversations to evolve dynamically, shifting topics based on user input rather than adhering to a structured task. Generative Artificial Intelligence (AI), which excels in producing fluid and contextually relevant responses, has significantly advanced open-domain conversational systems. While it can generate natural human-like conversations, it often lacks the precision, consistency, and structured reasoning required for goal-driven interactions. Without explicit task constraints, generative models may produce responses that fail to meet the critical requirements of task completion, factual consistency, and contextual coherence in IA systems.

Effectively handling textual and contextual information in task-oriented dialogues requires the IA system to possess **Natural Language Processing (NLP)** capabilities. Figure 1.1 illustrates the system components within TOD systems that are enabled by NLP techniques:

Conversational Input Processor (CIP): should be capable of recognizing and convert-

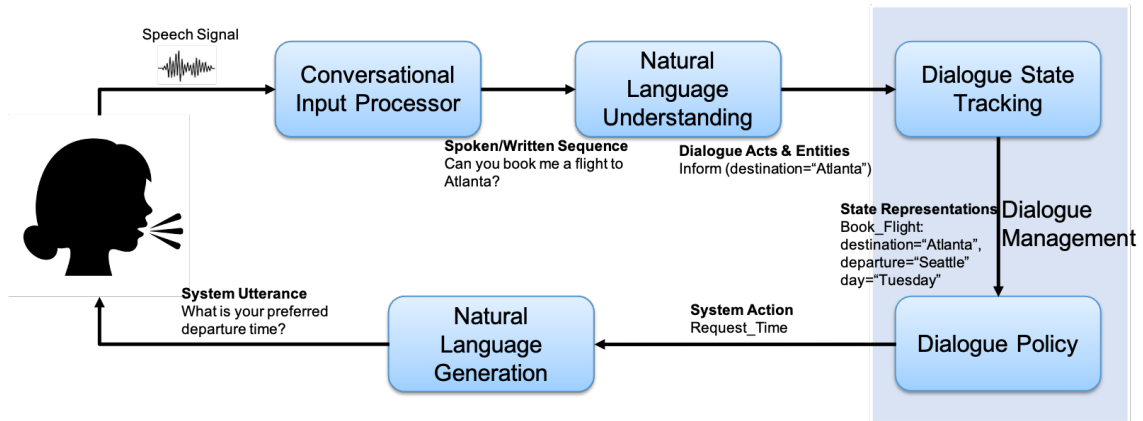


Figure 1.1: Pipeline of a task-oriented dialogue system

ing spoken language into text if the dialogue system supports verbal input. Moreover, CIP integrates NLP techniques, such as context understanding [4], error correction [5], and language modeling [6] to enhance accuracy and adaptability across diverse linguistic contexts and augment TOD systems with large-scale multilingual and multi-accent corpora.

Natural Language Understanding (NLU): parses user utterances into semantic frames that represent user actions. The system should accurately interpret user intent within the dialogue context and convert it into a format that the machine can process, such as structured natural language or a semantic frame. Part-of-speech tagging and named entity recognition are therefore crucial for the NLU tasks [7, 8].

Dialogue Management (DM): maintains the dialogue state and determines the system's next action. It processes past interactions, updates the current dialogue state, and generates appropriate responses. Dialogue management can be formulated as a semantic parsing task [9] or a machine reading comprehension problem [10], both of which are powered by advancements in NLP.

Natural Language Generation (NLG): generates natural language responses based on the DM's decision regarding the system's next action. The dialogue system must translate its internal actions into coherent and contextually appropriate utterances for the user. Language modeling plays a critical role in NLG, particularly in modern dialogue systems [11, 12, 13, 14].

Unlike open-domain systems, TOD systems are designed to guide conversations with a clear objective, ensuring task completion and accurate information retrieval. They require mechanisms such as dialogue state tracking, structured reasoning, and specialized knowledge integration to maintain contextual accuracy across multi-turn interactions. While advances have been made in NLP to enhance dialogue management [15, 16, 17] and response generation [18, 19, 20], TOD systems still face challenges related to knowledge integration, scalability, and generalization across multiple domains. Recent developments in Large Language Models (LLMs) have significantly impacted NLP applications, including conversational AI. LLMs, such as Llama [21, 22] and GPT-based models [23, 24, 25], exhibit remarkable proficiency in language comprehension and text generation. While LLMs have enhanced natural language processing capabilities in dialogue systems, their application in TOD remains challenging due to their lack of inherent structured memory and reliance on implicit knowledge rather than explicit task constraints. Therefore, integrating domain-specific and systematic knowledge remains crucial for advancing TOD systems beyond the capabilities of generative AI alone. Furthermore, task-oriented dialogue systems must process knowledge in various formats, ranging from unstructured text to user interaction histories, making it challenging to develop a unified framework for direct knowledge integration. In this thesis, we focus on non-database knowledge, which presents greater challenges due to its unstructured nature and its inherent complexity within domain-specific contexts. Leveraging this form of knowledge, we bypass the need for external databases, thereby enhancing the scalability and adaptability of the proposed approaches.

In summary, this thesis explores advances in NLP technologies to enhance dialogue generation, state tracking, and knowledge integration within TOD systems, advancing the capabilities of IA in providing more reliable, responsive, and contextually accurate interactions. It establishes methodologies for integrating *domain-specific*, *dialogue-level*, and *cross-lingual* knowledge to improve the system's ability to understand complex user goals, eliciting dialogue states effectively, and generate accurate, context-aware responses.

To achieve this, the proposed approaches address two major challenges: 1) formulating problems that dialogue systems can effectively handle with knowledge integration, and 2) using structured and unstructured knowledge to improve the scalability and adaptability of TOD systems across multiple domains. By incorporating diverse knowledge sources, including ontologies, unstructured text, user interaction histories, and model-generated outputs, this research ensures that the system can accurately interpret complex user intents, manage contextual dependencies, and generate adaptive responses to meet specific task requirements.

1.2 Motivation of Specialized Knowledge in TOD Systems

1.2.1 An Analogy of Human Expertise and Intelligent Assistants

Even highly educated professionals require domain-specific training to succeed in their respective industries. A computer science graduate, despite having a strong foundational knowledge, must learn industry-specific tools, programming frameworks, and business contexts to become effective in a particular role. For example, data scientists must master tools like Python, R, SQL, and machine learning frameworks. Game developers require expertise in game engines, computer graphics, and physics simulations. Similarly, IAs require fine-tuning to specialize in different domains. While they possess broad linguistic capabilities, their effectiveness improves significantly when trained on domain-specific datasets, such as legal terminology, medical diagnosis protocols, or business process automation, such as assisting developers by understanding and generating programming code [26], providing tutoring and assignment feedback [27], and supporting tasks like scriptwriting and storytelling [28]. These examples highlight that specialized knowledge is essential for both humans and IA in order to function optimally within a structured environment.

1.2.2 The Revolution of LLMs on TOD Research

Conventional TOD systems rely on machine learning models trained on specific datasets for tasks such as text classification, intent recognition, dialogue state tracking, and information retrieval. These models are domain-restricted and require separate training pipelines for different datasets, making them less adaptable to new domains without extensive retraining. LLMs have transformed this paradigm by unifying various text-processing tasks into a single framework. By formulating tasks as text-to-label generation problems, LLMs reduce the dependency on extensive annotated datasets while achieving performance comparable to, or even surpassing traditional models. This capability is largely attributed to their pre-training on vast corpora and their ability to interpret and follow structured instructions embedded within prompts.

Beyond classification and text generation, LLMs demonstrate advanced contextual reasoning, allowing them to identify implicit relationships within textual data and provide more insightful responses. Their ability to leverage parametric knowledge from pre-trained architectures enables them to expand text-based information, making them valuable for applications such as user modeling, recommendation, and personalization. For instance, research has shown that LLMs can effectively profile user preferences and detect sentiment by integrating behavioral history[29, 30]. Despite these advancements, LLMs alone remain insufficient for TOD systems, as they lack accurate reasoning mechanisms, goal-driven task execution capabilities, and direct access to domain-specific databases.

1.2.3 Why Task-Oriented Dialogue Systems Still Matter in the Age of LLMs?

Although LLMs excel at generating fluent and contextually rich responses, they are not inherently designed for structured, goal-driven interactions required in TOD systems. A well-functioning TOD system must not only understand user input but also actively guide the conversation toward task completion. This process involves maintaining contextual memory across multi-turn dialogues, dynamically tracking dialogue states, and proactively

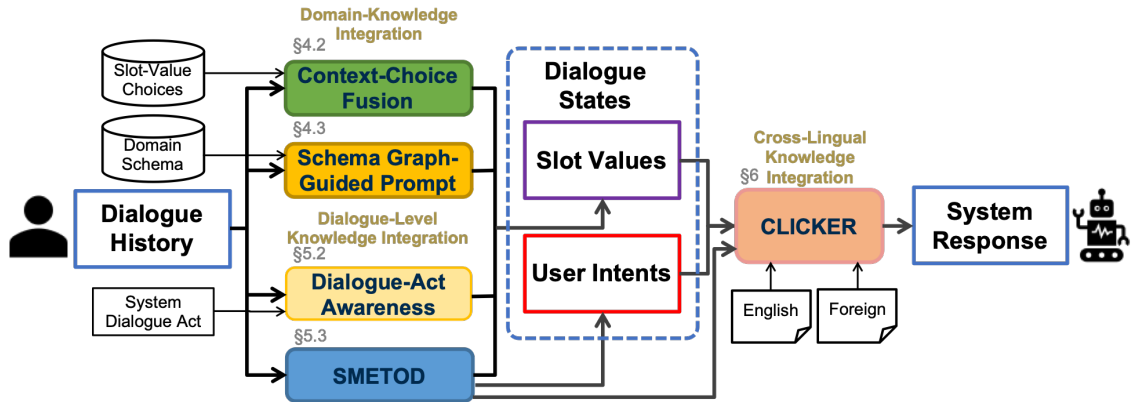


Figure 1.2: Illustration of the proposed knowledge integration approaches in this thesis.

requesting missing information when necessary [31].

A significant limitation of LLMs in TOD settings is their reliance on finite-context memory, which means they lack built-in mechanisms for persistent memory and structured task management and require external solutions such as dialogue state tracking or retrieval-augmented approaches. Moreover, while LLMs are designed for generalization across various tasks, they are not optimized for domain-specific constraints or operational requirements. Properly designed TOD systems incorporate rule-based logic, fine-tuned task-specific training, and integration with external databases to ensure high reliability and consistency. Recent research has sought to bridge the gap by leveraging LLMs as agents within multi-agent systems or by equipping them with tool-using capabilities for handling complex tasks. Studies have demonstrated that LLMs can enhance topic management in TOD systems [14], reduce hallucinations in generated responses [32], and improve user modeling by leveraging structured prompts [33, 34]. However, despite these improvements, TOD systems remain essential for structured interactions, as they provide a level of reliability, scalability, and task-specific knowledge that LLMs alone cannot achieve.

1.3 Thesis Outline

The following chapters provide an investigation of the key challenges in human-machine TOD systems, propose methodologies for knowledge incorporation, and present empirical

findings demonstrating the effectiveness of proposed approaches.

Chapter 2 introduces the background and challenges in existing TOD systems, emphasizing the role of structured knowledge in improving dialogue comprehension, enhancing task execution, and addressing limitations in scalability, generalization, and domain adaptation.

Chapter 3 presents methodologies for knowledge integration in TOD systems, which systematically captures user needs and defines knowledge at the domain, dialogue, and cross-lingual levels, and provides a structured approach to enriching TOD architectures.

Chapter 4 focuses on incorporating domain-specific knowledge into TOD systems. We propose methods for leveraging *structured slot-value* transfer[35] and *schema-guided knowledge graphs*[36] to improve DST, ensuring more accurate task execution and domain adaptation.

Chapter 5 explores dialogue-level knowledge integration in TOD systems, enhancing dialogue comprehension and response generation through *act-aware slot-value prediction*[37] and *mixture-of-experts*[38] techniques to ensure accuracy and efficiency in task-oriented dialogue systems.

Chapter 6 introduces cross-lingual knowledge transfer as a mechanism for enhancing Commonsense Reasoning (CSR) in low-resource languages[39]. By improving *commonsense knowledge* transfer across diverse language settings, it enhances the adaptability and effectiveness of TOD systems in multilingual environments.

Chapter 7 proposes a discussion of the key contributions, limitations of the study, and future research directions. It reflects on the broader implications of knowledge integration in TOD systems and outlines potential advancements in IA design.

As illustrated in Figure 1.2, the proposed methodologies are systematically integrated into TOD systems, demonstrating their role in improving dialogue capabilities. In summary, this work focuses on methodologies for integrating domain-specific, dialogue-level, and cross-lingual knowledge into dialogue systems, evaluating its impact across diverse

contexts to enhance the design of more effective and intelligent conversational agents. By enabling dialogue systems to incorporate knowledge on demand, it advances the development of more robust, scalable, and adaptable TOD systems.

CHAPTER 2

PRELIMINARIES IN DISCOURSE THEORY AND COMPUTATIONAL DIALOGUE SYSTEMS

2.1 Foundations of Computational Discourse Analysis

Customer service plays a crucial role in modern commercial activities, with interactions between users and computer systems now conducted almost entirely over digital communication networks. Early computational work in discourse primary focused on two forms: spoken and written forms of extended texts and dialogues [40]. Although there are clear overlaps that dialogues contain text-like sequences spoken by a single individual, and texts may incorporate dialogue, each of them presents distinct analytical challenges. Research on text is particularly relevant to document analysis and retrieval applications, while dialogue studies are important for human-computer interaction, regardless of modality. Despite their differences, text and dialogue share two key characteristics. *First*, discourses segment into meaningful units, much like sentences divide into phrases. Utterances are grouped into discourse segments, capturing meaning beyond individual components. *Second*, the interpretation of referring expressions is critical in both forms, influencing coherence and comprehension [41].

A primary focus of discourse research is understanding the information a discourse conveys beyond the literal meaning of individual sentences. [42] categorize discourse analysis into two broad approaches: *informational* and *intentional*. Informational approaches assert that discourse coherence arises from semantic relationships between successive utterances. In contrast, intentional approaches argue that coherence derives from the speaker's or writer's underlying intentions, with comprehension relying on recognizing these intentions. Later research underscores the significance of collaborative intentions and collective

actions in analyzing discourse[43].

A comprehensive dialogue analysis requires both domain-level representation—the semantic content of participants’ speech—and dialogue-level information, such as the types of speech acts¹ being performed [42]. Dialogue theories, developed as early as the 1980s, form the foundation for dialogue systems that identify relevant semantic and pragmatic features.

The organization and coherence of discourse have been studied extensively through various theoretical frameworks. [44] introduces a theory of discourse coherence, based on a small, limited set of coherence relations, applied recursively to discourse segments. Expanding on this, [45] proposes a tripartite framework for discourse structure, addressing distinct aspects of discourse through three interrelated components. Rhetorical Structure Theory (RST) [46], presents a hierarchical model of text organization, classifying spans as either nucleus (central) or satellites (supporting) elements within rhetorical relations. Building on hierarchical discourse structures, [47] proposed a schema-driven approach to discourse, ensuring coherence and guiding content selection during text generation. These developments illustrate the evolution of discourse analysis, from defining basic coherence relations to constructing structured frameworks for understanding and generating coherent texts.

2.1.1 Taxonomy of Communication Systems

Effective communication takes various forms, each characterized by distinct structures, goals, and interaction patterns. Understanding these forms is essential for analyzing both human-human and human-machine communication. In natural language interactions, communication can range from informal, spontaneous exchanges to structured, goal-driven dialogues. By categorizing these systems, we can better model how machines should process and generate human language. This taxonomy provides a foundation for designing con-

¹For example, asking a question, responding to a previous question, seeking clarification, or making a proposal.

versational AI systems, enabling them to engage in meaningful interactions by mimicking human communication patterns.

- **Conversation** refers to an informal exchange of thoughts, ideas, opinions, emotions, or information through spoken words among two or more participants [48]. Conversations are spontaneous and free-flowing, providing valuable insights into how machines might handle unstructured, multi-party interactions in natural language.
- **Dialogue** is a more structured and cooperative form of two-way communication between two participants. In contrast to conversation, it is goal-oriented and serves functions such as asking and answering questions, requesting information, building relationships, delivering messages, inducing actions, or resolving issues. Dialogue typically follows implicit rules, such as turn-taking, and provides a more formal exchange structure, making it a foundational model for designing human-machine interactions. Based on the argumentation theory [49], dialogue can be categorized into six fundamental types: inquiry, negotiation, information-seeking, deliberation, persuasion, and eristic dialogue.

2.1.2 Variability of Dialogue Intuitive

Task-oriented dialogue operates under a *pseudo-system initiative*, where the system responds to user intents based on a predefined dialogue policy. The system's primary role is to exchange and convey information to fill predefined slots. This differs from *mixed-initiative* dialogue, which explicitly supports a more natural and efficient interleaving of contributions from both users and automated services to collaboratively reach solutions [50]. Mixed-initiative dialogue employs a flexible interaction strategy where each agent contributes based on its strengths. Unlike fixed role assignments, agents negotiate their roles dynamically as the problem-solving process being solved. One participant may take the lead and guide the interaction while the other provides support at any time [51]. Integrating mixed-initiative dialogue introduces added complexity, as the system must seamlessly

adapt to human preferences, interaction styles, and problem-solving approaches. The dialogue agent should interact in a modality that is convenient for the human user, ensuring a smooth and efficient collaboration.

Task-oriented dialogues not being fully user-initiative is determined by the nature of service dialogues, as the agent also participates in the initiative at the same time. In order to solve problems and provide services efficiently, one should construct realistic, effective plans with users in the loop with an integrated view of the different aspects of reasoning about and using plans [50]. One of the common practices in dialogue system design is generating system dialogue acts as a premise for managing the generation of utterances to communicate with the user in turns. Based on the speech-act theory, responses to a speaker's utterances are guided by the hearer's recognition of the intentions underlying the utterances [52]. Those dialogue acts are activities that the system takes by giving the following utterance. Early dialogue acts are strictly regulated by domain expertise or discourse linguistics. For example, the dialogue act "*Request*" in an utterance typically represents an illocutionary action for soliciting information about the task.

2.1.3 Dialogue Acts

Dialogue acts extend previous computational speech act theories to include coordinated activities between agents in a conversation [53]. They model discourse as a collection of joint speaker-hearer actions, where meaning is constructed within the specific context of a conversation and grounded in mutual understanding between participants. To maintain coherence and structure, dialogue acts are categorized into four essential levels: core speech acts, argument acts, grounding acts, and turn-taking acts. A dialogue act represents the meaning of an utterance and serves as a classification system that tags utterances based on a combination of pragmatic, semantic, and syntactic criteria [54]. The detection of dialogue acts is useful for managing and leading to more natural and fluent dialogues. Dialogue act prediction can be performed at the surface level through unsupervised methods. Popular

approaches include using *n-grams* to model the probabilities of dialogue act sequences [55] or employing *word clustering* algorithms based on perplexity measures [54], providing a foundation for understanding and resolving tasks in dialogues.

In previous work, speech acts are taken as noisy observations of dialogue acts to update dialogue states with generative methods [56, 57] which represent dialogue states at each turn by modeling the joint probabilities in belief space. Such methods provide more reasoning for dialogue systems, while they are not robust to outliers and limit the flexibilities of dialogue models. More recent discriminative models have proved their capabilities in tracking dialogue states, by modeling DST as multi-task classification [58, 59, 60, 61] or question-answering problems [62, 63, 64, 65]. Recent discriminative DST models estimate user goals directly from the dialogue context, ignoring the NLU module and dialogue acts.

2.2 Computational Dialogue Systems

2.2.1 Overview of Dialogue Systems

A dialogue system regulates interactions between a system and a user by setting expectations for user input, guiding system responses, and controlling the flow of dialogue. Additionally, the dialogue structure and related information, such as domain and world knowledge, are modeled differently depending on the system’s architecture [66]. As [67] put it, dialogue systems can be categorized into three main types: *finite-state systems*, *form-based systems*, and *agent-based systems*.

From an application perspective, dialogue systems can be broadly classified as **open-domain dialogue systems** or **task-oriented dialogue systems**. **Open-domain dialogue systems** do not require specific responses or confirmations at each turn. One of the earliest examples of such a system is Eliza[68], developed in the 1960s. While some claimed that Eliza passed the Turing test, it primarily engaged users in casual, non-task-oriented exchanges, offering entertainment rather than functional dialogue. Other examples include Parry[69] in the 1970s, Alice[70] in the 2000s, and more recent systems such as Clever-

bot[71] and XiaoIce[72], which employ more sophisticated AI techniques for open-ended conversations. A real-world spoken dialogue example is Let's Go[73], designed to interact with users via a speech-based interface. Since open-domain conversation systems do not have explicit goals, their performance is often evaluated subjectively, based on factors such as user engagement and probability rather than task completion.

In contrast, most practical applications are limited to domain knowledge support, making **task-oriented dialogue systems** the dominant approach in dialogue system development. These systems assist users in completing specific tasks, such as answering questions within a prescribed domain or routing service requests to the appropriate department. In practice, a closed-domain dialogue system often incorporates multiple dialogue types to achieve a well-defined objective. These systems are designed to fulfill user expectations within a constrained scope, making them integral to business automation [74]. A task-oriented dialogue system must have a specific goal in mind, and recent advancements have expanded its applications across domains such as ticket booking, online shopping, travel planning, and recommendation systems. One of the earliest examples is the General Understanding System (GUS)[75], introduced in 1977 as a frame-based dialogue system for airline travel planning. Its structured approach has influenced modern commercial virtual assistants. Later study, such as GALAXY[76], introduced a client-server architecture for spoken dialogue systems that provide online information access. Similarly, ITSPOKE [77] employs a similar framework for automatic tutoring services using spoken dialogue.

The following sections will briefly review approaches to task-oriented dialogue systems, which are the primary focus of this thesis on dialogue system research and development.

2.2.2 Dialogue States

The dialogue state serves as a structured representation of constraints and relevant information exchanged during a conversation. It consists of predefined slots, each capable of

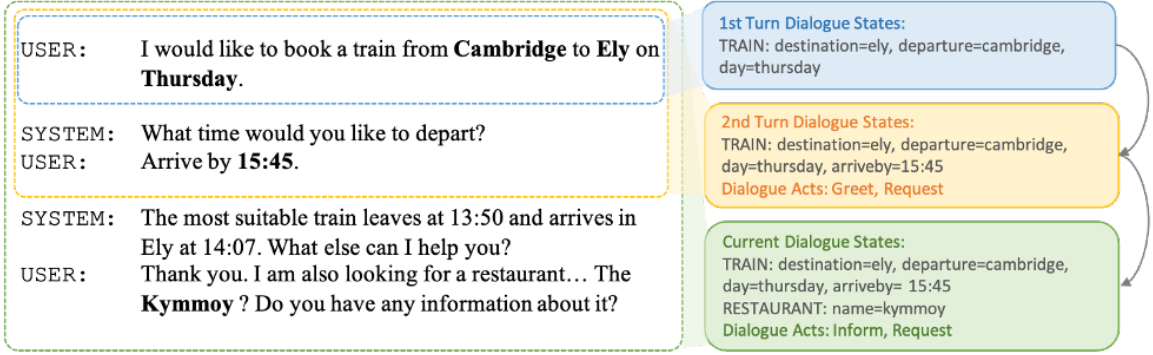


Figure 2.1: An example of a task-oriented dialogue with slot-value representations

holding a specific value. Figure 2.1 shows an example of a task-oriented service dialogue between a user and a system for services across train and hotel domains. At a given turn t , the objective of Dialogue State Tracking (DST) is to extract relevant values from the dialogue context while maintaining accumulated information from previous turns. The dialogue state, denoted as S_t , guides the system in determining the next action toward task completion, such as finding a suitable hotel.

Traditionally, dialogue states differentiate between *informable slots* and *requestable slots*. Informable slots represent constraints provided by the user (e.g., price range), while requestable slots contain information that the user can inquire about (e.g., phone number). Additionally, informable slots can take special values, such as *don't care* when the user has no preference and *none* when no goal has been specified for that slot. The dialogue state at turn t can be predicted either from the previous turn using S_{t-1} as context or from the entire dialogue history to infer updated states at each turn.

For single-domain dialogues, the dialogue state is typically represented as $(slot, value)$ pairs. However, when handling multi-domain conversations, this representation is extended to $(domain-slot, value)$ pairs, enabling the system to track multiple domains simultaneously. Task-oriented dialogue systems are often scenario-specific, assisting users in completing structured tasks such as hotel bookings or restaurant recommendations within a defined domain [78]. This contrasts with open-domain conversations found in corpora like Switchboard[79] and CALLHOME[80], which involve free-flowing, social interactions without

predefined goals [66].

Advances in DST research. Research on DST has evolved significantly, transitioning from generative models to discriminative models, and from fixed ontology-based methods to more scalable approaches for handling complex dialogues. Early methods formulated state transitions as probabilistic processes with fixed sets of dialogue states [57]. More recent approaches leverage slot-value pairs for dialogue state representation [81, 82], where slot values may either be drawn from predefined sets or extracted dynamically as free-form spans.

For predefined value sets, DST is often modeled as a classification problem, assuming fixed candidate values for each slot [58, 83, 84]. However, these methods struggle with unknown values and adapting to new domains due to their reliance on static ontologies. To overcome this limitation, TRADE[85] and Scalable Neural DST[86] introduced models capable of tracking slots with dynamically generated values. Inspired by advances in neural machine reading, recent work has shifted toward span-based DST, which bypasses the need for predefined value sets by extracting slot values directly from dialogue history.

Another promising direction involves leveraging Pre-trained Language Model (PLM) as a feature extractor, enhancing DST by incorporating schema-based representations and modeling relationships between different slots. SST[87] employs graph attention networks to represent schema graphs, allowing slots across domains to benefit from shared semantic representations. In contrast, ReDST[88] models dialogue state updates using a bipartite belief graph, where slot values propagate across turns based on prior beliefs and current utterances. These methods improve DST’s ability to handle complex slot dependencies and maintain inter-utterance consistency.

Through these advancements, DST has evolved from rigid ontology-driven frameworks to more flexible, scalable models, paving the way for robust, adaptive dialogue systems in real-world applications.

Scalable Dialogue State Tracking. Scaling DST for real-world applications presents sig-

nificant challenges due to the high complexity of domain, slot, and value combinations. At every dialogue turn, DST models must evaluate all possible slot-value candidates to predict the current dialogue state, leading to increased computational demands as the system scales. While data-driven approaches to dialogue systems have been developed to handle tasks across multiple application domains, they often fail to address key challenges associated with scaling virtual assistants in production [82].

Deploying task-oriented dialogue systems typically requires extensive annotated DST datasets, which are both time-consuming and expensive to collect for every new domain and service. Despite advances such as the copy mechanism [58, 89, 90] and dialogue act-based approaches [63, 62], which enhance slot and value tracking within dialogue histories, DST performance still heavily depends on large amounts of annotated data. This reliance poses significant barriers to scaling, especially for tasks that require adapting to new domains or services efficiently.

To address these inefficiencies, several methods have been proposed to improve the scalability and computational efficiency of DST systems. For example, [91] and [92] introduced decoders that generate dialogue states as sequences rather than slot-value pairs, reducing the computational overhead associated with candidate evaluation. Additionally, single-pass approaches [93, 82, 94] encode schema embeddings only once, significantly saving computational resources.

2.2.3 Finite state-based systems

A finite state-based system structures dialogue as a sequence of predefined states, with the system guiding the user through a fixed dialogue graph. Each state corresponds to a specific prompt, and based on the user’s response, the system executes an action and transitions to the next state. This rigid structure ensures simplicity, reliability, and low error rates, making it ideal for well-defined tasks such as booking services or schedule inquiries [67]. However, its lack of flexibility makes it unsuitable for complex interactions requiring negotiation or

dynamic adaptation [66].

2.2.4 Form-based systems

A form-based system structures dialogue as an information-gathering process, where the system dynamically fills slots in a structured form based on user input. Unlike finite state-based systems, form-based dialogue flows are determined dynamically, allowing users to take initiative and provide additional details. This flexibility improves efficiency but requires more complex dialogue management. While form-based systems are well-suited for structured tasks, they struggle with highly dynamic domains that require deeper contextual reasoning.

2.2.5 Agent-based systems

An agent-based system models dialogue as a collaboration between intelligent agents, leveraging AI techniques to reason about goals, beliefs, and intentions [95]. Unlike rule-based approaches, these systems dynamically adapt to user input, allowing topic shifts, complex decision-making, and proactive system responses. While this flexibility makes agent-based systems ideal for interactive tutoring, planning, and virtual assistants, their high computational demands limit their scalability and implementation feasibility in many real-world applications [96].

2.2.6 Challenges in Task-Oriented Dialogue Systems

TOD systems primarily assist users in completing specific tasks through natural language interactions [18]. Considering a broad range of applications, task-oriented dialogue systems should generate diverse types of outputs for processing information, evaluating user intentions, or retaining for future reference. In real-world scenarios, processed dialogue information can be structured in different formats, such as form-based[15], probability-based[97], or text-based representations [13].

This leads to two predominant system designs, namely *pipeline-based* and *end-to-end*. Pipeline-based systems process dialogue in modular stages, but they suffer from error propagation, where mistakes in one module affect subsequent components [98]. End-to-end systems, which learn to map inputs directly to outputs, mitigate this issue but can struggle with missing essential information beyond just generating responses. Additionally, debugging and interpreting model decisions is often more challenging in end-to-end approaches [99].

Pre-trained task-oriented dialogue models have recently outperformed traditional fine-tuning approaches, demonstrating strong generalization across multi-objective dialogue scenarios [11, 100, 101, 102]. However, these models require large-scale dialogue data for training and lack an efficient interface for optimizing sub-modules, making real-world deployment challenging.

Another major constraint in scaling dialogue models is their computational efficiency. While PLMs have advanced dialogue systems through transfer learning [103, 104, 23, 105], their growing size increases computational costs. To address this, parameter-efficient adapters have been introduced, allowing only a small number of parameters to be updated while keeping the PLM frozen [106, 107, 108]. These approaches have gained popularity in dialogue systems [99, 109], but they remain computationally expensive due to their sequential nature [110]. Furthermore, inference time scales with model complexity, making real-time deployment increasingly challenging. Addressing these efficiency concerns is crucial for advancing scalable, high-performance task-oriented dialogue systems.

2.3 Knowledge in TOD Systems

2.3.1 Domain-Specific Knowledge

Domain-specific knowledge refers to the syntactic and semantic relationships within a particular field, providing a structured representation by mapping user input to slots and values, helping constrain the search space and improve efficiency. It allows the system to extract structured information, such as intents, entities, and slot-value pairs, which help the system

the system understand user precisely and complete task efficiently.

Previous research has explored various techniques for integrating domain knowledge into DST, including slot-value relationships through linked graphs[62], belief propagation on bipartite graphs[88], and ontology-aware attention mechanisms[87], and combining prior knowledge with dialogue context through schema graph evolution [111]. Recent advancements enhance domain-specific knowledge extraction. Techniques such as schema descriptions and natural language-defined slot values have been used to condition PLMs for improved slot-filling and state tracking [112, 113, 114]. Additionally, methods that prepend task-related examples to input prompts help refine dialogue state predictions by conditioning the model on relevant contexts [115, 116, 117]. However, they rely heavily on large labeled datasets and may fail to generalize to complex schema representations.

In **chapter 4.2**, our work addresses this challenge by transferring slot-value contextual knowledge from a question-answering dataset. **chapter 4.3** introduces a schema-guided approach that further enhances generalization by jointly encoding the dialogue context and domain schema into a unified representation space.

2.3.2 Dialogue-Level Knowledge

2.3.3 Cross-lingual Knowledge

Cross-lingual knowledge transfer enables multilingual dialogue systems to generalize across languages. One common approach is Neural Machine Translation (NMT), which converts text from various languages into English, leveraging the natural language understanding capabilities of English-trained models [118, 119, 120, 121]. Additionally, multilingual representations have been explored for knowledge transfer, integrating task-specific architectures to improve system performance across languages [122, 123, 124].

Pre-trained multilingual language models have demonstrated significant potential for cross-lingual knowledge transfer, excelling in various evaluation tasks [24, 125, 126, 127, 128]. However, performance discrepancies remain between English and other languages,

particularly in cross-lingual conversational speech recognition. To address this, our work in **chapter 6** focuses on minimizing these cross-lingual discrepancies, improving the system’s ability to handle diverse languages with consistent accuracy.

2.4 Using Large Language Models in Dialogue Systems

2.4.1 Applications of Large-Language Models

Large Language Model (LLM)s are increasingly integrated into dialogue systems to enhance user experience by leveraging behavioral history and preferences for personalized interactions. Later work [129] utilized LLMs to retrieve user behavior context and history for personalized response generation. DiagGPT[14] employs a multi-agent framework, where LLMs track conversation topics, ask relevant questions, and guide users toward task completion. RefGPT[32] aims to generate truthful and customized dialogues while minimizing factual errors caused by model hallucination. Additionally, LLMs have been applied to synthesize user behavior data to improve dialogue modeling [33, 34]. These developments highlight the evolving role of LLMs in enhancing the functionality of TOD systems, while emphasizing the continued importance of TOD system.

2.4.2 Understanding Complex Data Structures Using LLMs

Using structural relations between in-domain attributes is an effective way to construct and distill features. GNN has been successfully applied in various dialogue applications. For example, [130] adopts a graph convolutional network (GCN) for utterance-level emotion recognition. [131] modeled structured dialogue policy with GNN; [132] and [133] propose a joint framework leveraging graph attention network [134] for both dialogue act recognition and sentiment classification. Recently, DSGFNet [111] leverages the schema graph encoder that fuses slot-domain and dialogue-aware relations for DST; DiCoS-DST [135] selects relevant turns dynamically by modeling structural slot dependency.

2.4.3 Domain Adaptation Using LLMs

Prompting is the approach of adding extra information for the model to condition when generating the output. As LLMs grow larger and larger, it becomes increasingly infeasible to perform conventional fine-tuning, where separate sets of the model parameters are modified per single task. To reduce the effort in fine-tuning large LLMs and promote the scalability of domain adaptation, there is a line of work that fixes the entire LLMs and introduces a small number of new trainable parameters. Notable examples in this category include adapter [106, 136, 137], prefix-tuning [107] and prompt-tuning [108], *etc.* Different from prompt design, which requires manual selection of prompt tokens from a fixed vocabulary, prompt tuning maintains a fixed set of special tokens, whereas only the embeddings of these prompt tokens are tuned. Prompt tuning performs comparably or even better with conventional full-model tuning when downstream data are sufficient. Further studies improve prompt tuning by initializing it with an extensive training stage [138, 139]. In DST, prompts are used to help knowledge transfer in continual learning [140, 141] or address the capacity issue of domain adaptation [112]. In end-to-end dialogue systems, a line of work prompts with specific text to generate desired outputs [142] or injecting adapters to capture the knowledge of different functionalities [99, 143]. GPT-3 [144] and ChatGPT² are also successful and efficient open-domain dialogue systems. On the other hand, the MoE approach focuses on improving performance by efficiently scaling model sizes. On the other hand, the Mixture-of-Experts (MoE) approach focuses on improving performance by efficiently scaling model sizes, which is applied for extracting dialogue-level knowledge in **chapter 5**. Recent work on MoE develops more efficient routing implementations of Mixture-of-Experts in scaling language models [145, 146, 147, 148].

²<https://chat.openai.com/chat>

CHAPTER 3

METHODOLOGY FOR KNOWLEDGE INTEGRATION IN TASK-ORIENTED DIALOGUE SYSTEMS

For a dialogue system to be effective across various conversational scenarios within a domain, its structure must be generalizable to accommodate different user intents and responses. Task-oriented dialogue systems typically involve progressive slot-filling, where users and the system collaboratively supply, verify, and refine information. As the conversation goes, relevant slots in the dialogue state are iteratively populated. Once all required slots are filled, the system executes an associated action, such as booking a hotel or confirming a reservation. The system execution can be a function calling or a sequential response. The success of goal-directed interactions depends on maintaining both accuracy and consistency in the collected information. This section outlines our methodology using knowledge into various stages of TOD systems, as well as knowledge transfer approaches that enable domain and task adaptation.

3.1 Proposed Methodologies

Our methods assume that dialogue understanding and generation depend on three forms of knowledge:

3.1.1 Domain-Specific Knowledge

Domain-Specific Knowledge (DSK) provides the structural and content-specific elements necessary for interpreting and processing user input in task-oriented dialogue systems. It maps user expressions to structured representations such as slots and value, which is defined by semantic rules. Domain-specific knowledge defines the scope and constraints of a service domain, including slot definitions, value constraints, and relationships within the

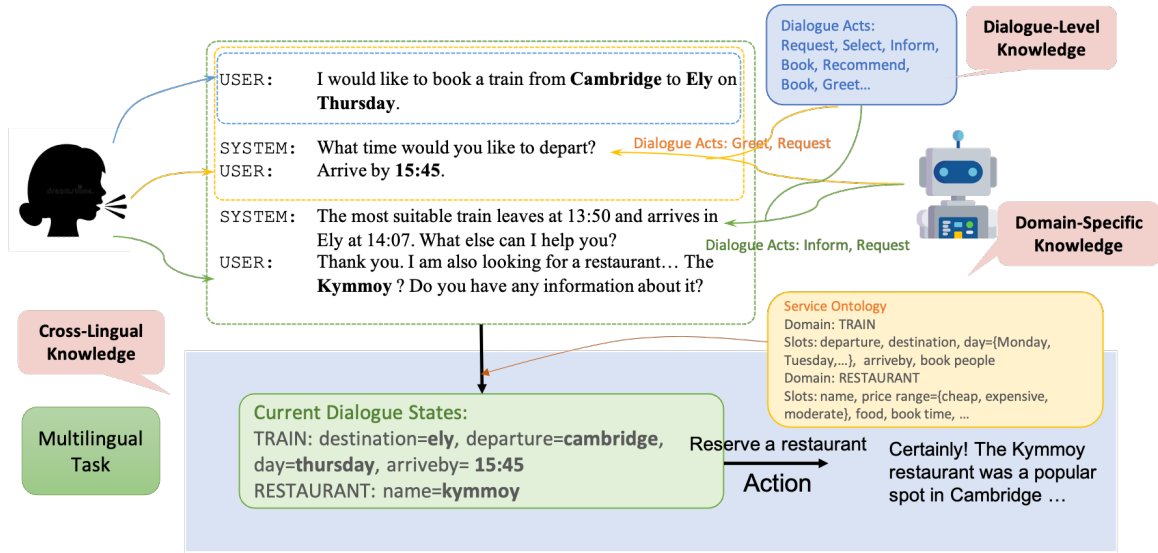


Figure 3.1: Methodology for intervening dialogue state tracking at different levels.

domain schema. This knowledge is essential for DST, enabling models to interpret user intent accurately and adapt across domains. Structural relations between in-domain attributes help extract meaningful features, making DST more scalable and interpretable.

Domain-specific knowledge focuses on mapping user input (\mathcal{U}) to structured representations (\mathcal{C}) using domain-specific rules (\mathcal{D}_s).

$$\mathcal{K}_s : \mathcal{U} \times \mathcal{D}_s \rightarrow \mathcal{C} \quad (3.1)$$

Where $\mathcal{U} = \{u_1, \dots, u_n\}$ represents the user input sequence, such as tokens or utterances. \mathcal{D}_s denotes domain-specific knowledge, including ontologies, schemas, or slot-value structures. $\mathcal{C} = \{(s_1, v_1), \dots, (s_m, v_m)\}$ is the structured representation, where each slot s_i (e.g., "departure") is paired with its corresponding value v_i (e.g., "Cambridge"). This formulation ensures that domain-specific knowledge effectively structure and interpret user input, facilitating accurate and meaningful dialogue interactions.

3.1.2 Dialogue-Level Knowledge

Dialogue-Level Knowledge (DLK) captures the contextual and pragmatic aspects of communication, focusing on speaker intentions, implied meanings, and interactive dynamics. While domain-specific knowledge deals with content interpretation, dialogue-level knowledge emphasizes goal-driven interactions, ensuring that dialogues remain coherent and responsive[149].

Dialogue-Level knowledge can be derived from preceding modules and models within the dialogue system, such as dialogue acts, making it essential for TOD systems. Different dialogue architectures incorporate knowledge in various ways. State-based architectures follow a fixed sequence of information retrieval, ensuring structured interactions suited for well-defined tasks. Form-based architectures store dialogue history in forms and slots, allowing flexibility in the dialogue flow while remaining decoupled from a control mechanism. In contrast, agent-based architectures require advanced NLP techniques to infer beliefs, goals, and intentions, which are often implicit rather than explicitly stated in the dialogue. These systems demand higher adaptability to handle complex, evolving user interactions.

Dialogue-level knowledge defines how contextual rule \mathcal{D}_p influences the dialogue interaction and shapes conversation flow. This process can be formally defined as:

$$\mathcal{K}_p : \mathcal{U} \times \mathcal{D}_p \rightarrow \mathcal{R} \quad (3.2)$$

Where $\mathcal{U} = \{u_1, \dots, u_n\}$ represents the dialogue history, which is the user input sequence as before. \mathcal{D}_p is domain-independent contextual rules, including system dialogue act types and user intents derived from NLU models. \mathcal{R} represents either a set of dialogue state pairs or a sequence of response tokens generated by the dialogue system. Specifically, system dialogue acts can be predicted through unsupervised learning and utilized for tracking dialogue states. In contrast, user intentions can be inferred and incorporated as part of

the input context to enhance subsequent dialogue state generation. This formulation captures the role of dialogue-level knowledge in facilitating contextually appropriate dialogue management and response.

3.1.3 Cross-Lingual Knowledge

Cross-Lingual Knowledge (CLK) enhances a dialogue system’s ability to generalize across languages, allowing models to transfer linguistic and commonsense reasoning capabilities beyond English-based training data. By incorporating multilingual representations and transfer learning techniques, task-oriented dialogue systems can adapt to new linguistic environments while maintaining their ability to process structured dialogue information.

Cross-lingual knowledge transfer in TOD systems enables models to generalize linguistic and commonsense reasoning capabilities across multiple languages. It ensures that knowledge learned in a source language (i.e., English) can be effectively transferred to a target language, maintaining coherence, reasoning accuracy, and task efficiency. This process can be defined as follows:

$$\mathcal{K}_x : (\mathcal{M}_s, \mathcal{D}_x) \rightarrow (\mathcal{M}_t, \mathcal{D}_x) \quad (3.3)$$

The notation \mathcal{D}_x represents commonsense knowledge, which is assumed to be shared across languages. \mathcal{M}_s and \mathcal{M}_t correspond to multilingual language models specific to the source and target languages, respectively. Our objective is to train \mathcal{M}_t to effectively perform target language reasoning while leveraging commonsense knowledge across languages.

3.1.4 Applications

Following the above framework of knowledge incorporation, the approaches proposed in this thesis can be categorized into three areas:

- **Incorporating domain-specific knowledge into task-oriented dialogue systems.**

Domain-specific knowledge provides the structural foundation for interpreting user input within a constrained domain. It defines slot-value mappings, ontologies, and schema relationships, ensuring accurate DST. To enhance system generalization, our work introduces methods that leverage domain-level knowledge to improve DST in new domains.

One such approach involves **domain-specific slot-value** transfer, where domain-agnostic QA datasets are used to mitigate the dependency on annotated in-domain data. Instead of requiring extensive labeled dialogue samples, this model transfers knowledge by directly utilizing candidate choices as domain constraints.

Another contribution integrates **schema-guided knowledge graph** into DST, incorporating slot relationships through a structured schema representation. By learning domain-aware prompts from schema graphs, this approach enables the system to adapt more effectively to new domains.

- **Incorporating dialogue-level knowledge into task-oriented dialogue systems.**

Dialogue-level knowledge involves the broader structure of dialogue interactions, guiding system behavior through contextual rules, dialogue acts, and modular architectures. By capturing the implicit logic of conversations, this knowledge layer facilitates more coherent and contextually appropriate interactions.

One approach that incorporates **system dialogue acts** into DST, which enhances DST by leveraging dialogue acts to predict categorical and non-categorical slot values. Using machine MRC techniques, the model systematically integrates dialogue act annotations, improving interpretability while ensuring adaptability to multi-domain settings. The study demonstrates how incorporating structured dialogue acts refines state tracking, leading to improved system performance.

Another approach enhances the scalability of task-oriented dialogue systems leveraging **retrieved information by dialogue states**. Traditional pre-trained language

PLMs face inherent capacity limitations, making scaling costly and computationally intensive. To address these challenges, this approach introduces a modular framework where multiple expert models specialize in different subproblems. By dynamically selecting and combining expert models, the system maintains high task-solving accuracy while optimizing computational efficiency. This modular architecture allows TODs to scale flexibly without compromising inference efficiency.

- **Enhancing understanding across languages.**

The expansion of dialogue systems to multilingual settings introduces challenges related to commonsense reasoning and knowledge transfer. Since large-scale language models are often trained on English-centric datasets, cross-lingual discrepancies arise, limiting system performance in low-resource languages.

To address these challenges, our work proposes **cross-lingual commonsense transfer**, a three-step framework based on XLM-R that enhances commonsense reasoning in multiple languages. By distinguishing between language-specific and commonsense knowledge, this method effectively bridges the performance gap between English and non-English language models. The framework improves foreign-language performance on commonsense reasoning benchmarks, demonstrating its effectiveness in cross-lingual knowledge adaptation. This contribution aligns with the need for robust multilingual understanding, where knowledge is transferred across languages while preserving reasoning capabilities.

3.2 Conclusion

The structured integration of domain-specific, dialogue-level, and cross-lingual knowledge plays a fundamental role in advancing task-oriented dialogue systems. The incorporation of domain-specific knowledge improves zero-shot DST and schema-guided prompt learning, ensuring the system can generalize to new domains with minimal supervision. Dialogue-

level knowledge enhances response accuracy and system scalability, leveraging structured dialogue acts and modular architectures. Enhancing multilingual understanding through cross-lingual commonsense transfer ensures that dialogue systems remain effective across diverse linguistic contexts. Together, these advancements contribute to the development of robust, scalable, and adaptable TODs capable of operating across multiple domains and languages.

CHAPTER 4

INCORPORATING DOMAIN-SPECIFIC KNOWLEDGE FOR TASK-ORIENTED DIALOGUE SYSTEM

4.1 Introduction

Effective TOD systems enable users to express beliefs, goals, and intentions, facilitating collaboration and problem-solving beyond individual capabilities. However, dialogue systems extends beyond simply processing and generating utterances; it requires contextual reasoning, structured knowledge integration, and adaptation to dynamic user interactions.

This chapter explores methods for leveraging domain-specific knowledge to enhance DST while reducing dependency on annotations. The fundamental principle behind knowledge transfer in DST stems from the observation that dialogue understanding shares common patterns across domains.

Our work on slot-value transfer represents a significant advancement in this direction, which reformulates DST as a Question Answering (QA) task. Instead of relying on predefined slot-filling mechanisms, the system generates natural language questions for each slot in the dialogue state and extracts relevant answers from the dialogue context. This reformulation enables DST models to benefit from the semantic richness of large-scale QA systems, improving adaptability across domains.

We propose a schema-guided approach that further enhances generalization by encoding both the dialogue context and domain schema into a unified representation space. This method enables the model to recognize relationships between user utterances and domain-specific concepts without requiring extensive training for each new domain. By using schema information as a bridge, the system can effectively transfer knowledge between domains, allowing for scalable and adaptable task-oriented dialogue systems.

4.2 Dialogue State Tracking with Slot-Value Transfer

4.2.1 Motivation

Nowadays, the requirements of deploying a number of services across multiple domains raise challenges to DST models in production [82]. However, existing dialogue datasets only span a few domains, making it impossible to train a DST model upon all conceivable conversation flows [150]. Furthermore, dialogue systems are required to infer dialogue states with dynamic techniques and offer diverse interfaces for different services, adding complexity to their implementation and deployment. Despite the fact that the copy mechanism [58, 89, 90] or dialogue acts [37, 63, 62] are leveraged to efficiently track slots and values in the dialogue history, the performance of DST still relies on a large number of annotations of dialogue states, which is expensive and inefficient to collect data for every new domain and service.

The traditional supervised learning approach, which relies heavily on manual annotations for each domain, has become increasingly impractical. The process of collecting and annotating domain-specific data is both expensive and time-consuming, creating a bottleneck in the development and deployment of new dialogue systems. This limitation has sparked interest in alternative learning approaches, particularly zero-shot and few-shot learning methods, which aim to leverage neural networks for greater system flexibility and adaptability. Learning with little or no new annotated data offers an alternative to taking advantage of the capabilities of neural networks to guarantee the flexibility of the systems. The importance of this aspect is reflected by recent works that address this problem [65, 151].

Data collection and training costs limit the performance of DST on a large scale. A line of work suggests jointly encoding the schema and the dialogue context for DST to address the above challenge [152, 112]. On the other hand, based on the language processing theory [153]—meanings in similar context could be understood and predicted before

encountered, large-scale QA datasets provide an option to transfer learned knowledge for DST with little to no in-domain data without the loss of performance. [151, 154]. However, none of the existing works model candidate choices in DST explicitly, resulting in a lack of efficiency and interpretability in using knowledge extraction capabilities from the QA datasets.

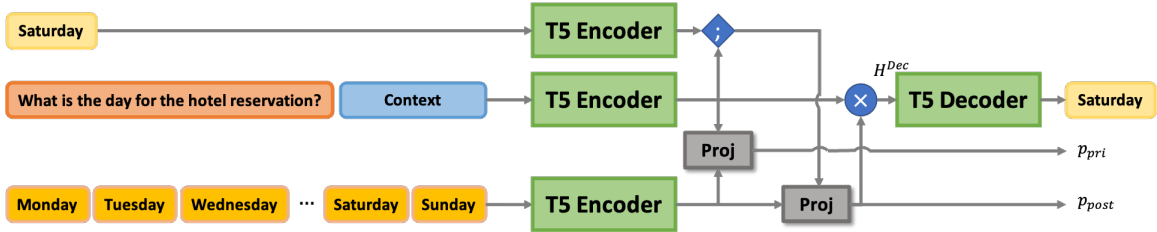


Figure 4.1: The overview of **Choice-Fusion Dialogue State Tracking (CoFunDST)**. The input is a concatenated query consisting of a question (Tangerine) and reference context (Blue). A sequence of choice tokens (Orange) is used for training and reference. The golden answer token (Yellow) is an extra input during training as extra supervision for encoding the concatenated query.

As a means of tackling these issues, we propose **Choice-Fusion Dialogue State Tracking (CoFunDST)** [35] that trains on extensive QA datasets with sufficient annotations for zero-shot DST without training on particular domains¹. Specifically, it fuses candidate choices as knowledge to predict slot-values accurately based on a T5 [155] pre-trained encoder-decoder language model. CoFunDST formulates DST and QA as machine reading comprehension (MRC) [156, 157] which generates the answer given the reference context. As a part of CoFunDST, we design appreciative choice selection to assess the relevance of all available candidate choices to the reference context and compute a probability distribution over these choices. Then we apply context-choice fusion [158] to incorporate the context-dependent choices as knowledge for initializing the decoder. Our work advances zero-shot DST in the following ways:

- For the first time, we model candidate choice of values as a distinctive resource of knowledge to leverage missing details for predicting slot-values accurately.

¹Code is publicly available at <https://github.com/youlandasu/Choice-Fusion>.

- We propose the context-choice fusion to selectively incorporate encoded choices based on the dialogue context.
- The performance of our model is demonstrated on the MultiWOZ 2.1 [159] showing it outperforms existing zero-shot DST approaches in terms of joint goal accuracy in “Restaurant”, “Train”, and “Taxi” domains. Further analysis shows its effectiveness of the choice fusion and knowledge transfer from QA to DST generating different types of slot-values.

4.2.2 Method

Problem Formulation.

Both QA and DST are formulated as generative MRC problems, which take questions and choices as input and generate answers token-by-token by comprehending the reference context, as depicted in Figure 4.1. For QA training, the input query combines the sequence of question tokens, i.e. $q = \{q_1, q_2, \dots, q_K\}$, and the reference context tokens, i.e. $c = \{c_1, c_2, \dots, c_L\}$, of the length K and L , respectively. In other words, the model can be regarded as filling the sequence a with correct tokens given the question q and the reference context c : “*question: q context: c answer: [a]*”, where $a = \{a_1, a_2, \dots, a_M\}$ and M is the length of answer. Additionally, the concatenation of N candidate choices of the given question is denoted as $v = \{v_1, v_2, \dots, v_N\}$. We also encode the ground truth answer \tilde{a} in the training set into tokens to combine it with the encoded input.

For DST reference, each domain slot is re-formulated as a natural language question in the form of “*What is the [slot] of [domain] that the user is interested in?*”, or “*What time*” and “*How many*” as prefixes of time- and number-related slots, similar to the domain-slot formulation in [151]. In particular, the state value and dialogue context are taken as an answer and reference context, respectively.

Choice-Fusion Mechanism

Appreciative Choice Selection: The appreciative choice selection is designed to select the choices that are highly relevant to the reference context, i.e. the *appreciative choices*. The choice tokens v are processed by the T5 encoder as $V \in \mathbb{R}^{N \times T}$, where T is the output hidden dimension of the Transformer. We calculate the *prior* and *posterior* probability distributions, p_{pri} and p_{post} , of the candidate choices V given the encoded question-context concatenation $D_{pri} \in \mathbb{R}^{(K+L) \times T}$ and the encoded golden answer $D_{post} \in \mathbb{R}^{M \times T}$. Note that only the prior distribution of Eq. Equation 4.1 is used during reference.

$$p_{pri} = \text{softmax}(\tanh(VW^V) \tanh([D_{pri}]W_{pri}^D)) \quad (4.1)$$

$$p_{post} = \text{softmax}(\tanh(VW^V) \tanh([D_{pri}; D_{post}]W_{post}^D)) \quad (4.2)$$

where $W_{pri}^D \in \mathbb{R}^{T \times F}$, $W_{post}^D \in \mathbb{R}^{2T \times F}$, and $W^V \in \mathbb{R}^{T \times F}$ are trainable parameter matrices, F is the intermediate dimension for projecting context on the choices. Then the objective function is the Kullback–Leibler divergence (KLD) [160] to optimize the distance of *prior* and *posterior* distributions of V , where the ground truth answer embedding D_{post} is served as the posterior knowledge for choice selection (Eq. Equation 4.3).

$$\mathcal{L}_{KLD} = KLD(p_{pri}, p_{post}) \quad (4.3)$$

Context-Choice Fusion: The context-choice fusion leverages appreciative choices to address choice knowledge in answer generation, by fusing the context and appreciative choices to the decoder. To fuse the obtained appreciation over choices for generating accurate answers, candidate choices are weighted by the posterior distribution p_{post} to initialize the input of the decoder, as shown in Eq. Equation 4.4.

$$H^{Dec} = \tanh([D_{pri}; p_{post}^T V] \cdot W^{Dec}) \quad (4.4)$$

where $H^{Dec} \in \mathbb{R}^{(K+L+N) \times T}$ is the fused input to the decoder, and $W^{Dec} \in \mathbb{R}^{T \times T}$. For inference, the prior distribution p_{pri} is used in replace of p_{post} for Eq. Equation 4.4. Such that the appreciative choices are contextualized and incorporated into the CoFunDST model as knowledge.

The overall objective in Eq. Equation 4.5 is the sum of the KLD and the Cross-Entropy loss of the decoder output and the ground truth answer \tilde{a} , where non-categorical slots and categorical slots are jointly trained.

$$\mathcal{L} = -\log P(a = \tilde{a} | D_{pri}, D_{post}, V) + \mathcal{L}_{KLD} \quad (4.5)$$

4.2.3 Experiments

Datasets: The model is pre-trained on 20% of the combination of six extractive QA datasets and two multi-choice QA datasets following the dataset pre-processing and slicing in [151]. To verify the generalization among different domains, we evaluate models on MultiWOZ 2.1 [159] and follow dataset setups in [58]. There are 30 distinguished domain-slots in MultiWOZ 2.1 in total, where there are 12 categorical slots provided with collections of values and 18 non-categorical slots.

Baselines: We select the following models as zero-shot DST baselines. (1) TRADE [58] is an encoder-decoder model which leverages slot gates and copy mechanism and shares parameters for predicting unseen slot-values. (2) SUMBT [161] uses pre-trained BERT [103] to learn the relations between slot types and values appearing in utterances and predict dialogue states with slot-utterance matching. (3) MA-DST [162] encodes dialogue context and domain-slots with attention mechanisms at multiple granularities to learn at different semantic levels. (4) TransferQA [151] proposes a task-transfer framework and takes the combination of slot, values, and dialogue context as the input for zero-shot DST.

TRADE [58], MA-DST [162], and SUMBT [161] are evaluated in the cross-domain setting, where the models are trained on the four domains in MultiWOZ 2.1 and evaluated

Table 4.1: Zero-shot joint goal accuracy on MultiWoz 2.1 [159]. Model results of TRADE [58], MA-DST [162], and SUMBT [161] are obtained from reference papers. TransferQA (T5-Small) follows all setups in [151], but is trained using T5-Small.

Model	Joint Goal Accuracy				
	Hotel	Train	Restaurant	Attraction	Taxi
TRADE [58]	14.20	22.39	12.59	20.06	59.21
MA-DST [162]	16.28	22.76	13.56	22.46	59.27
SUMBT [161]	19.80	22.50	16.50	22.60	59.50
TransferQA (T5-Small) [151]	21.82	25.66	17.98	26.14	59.68
CoFunDST (T5-Small)	21.07	25.95	18.13	24.79	60.19

on the held-out domain. On the contrary, TransferQA [151] and our model are trained on the combined QA dataset only. Therefore, it is unnecessary to use in-domain DST data. Although TransferQA and ours are both based on domain-agnostic QA training, we extend this idea with knowledge fusion and directly use values in domain ontology as candidate choices to align predictions of DST with the training procedure.

Implementation: We implemented our model based on T5-Small [155], which is a pre-trained encoder-decoder model for natural language generation. The input text of the T5 encoder is truncated to 512 tokens. The intermediate dimension for prior and posterior probabilities is 64. The fused input to the T5 decoder is passed through a 512-unit Feed-Forward layer for initialization. Adafactor [163] is used as the optimizer with initial learning rate $3e-4$ and warm-up steps 100. The number of training epochs is 6. As for our implementation of TransferQA [151], all other hyper-parameters are kept the same as the original settings except for the model size.

4.2.4 Results

The zero-shot joint goal accuracy (**JGA**), which is the average accuracy of predicting all slot-values of a turn correctly, on MultiWOZ 2.1 is presented in Table 4.1. It can be seen that our model exceeds all baselines on JGA in “*restaurant*”, “*taxi*”, and “*train*” domains with KLD loss and context-choice fusion adopted. Our inputs are shortened compared to

Table 4.2: Ablation study on the two components of the choice-fusion mechanism: appreciative choice selection with KLD loss (KLD) and context-choice fusion (Fuse). T5-Small [155] is used and the evaluation results on joint goal accuracy (JGA), slot goal accuracy (SGA), and F1 in five domains of MultiWoz 2.1.

Settings	Hotel			Train			Restaurant			Attraction			Taxi		
	JGA	SGA	F1	JGA	SGA	F1	JGA	SGA	F1	JGA	SGA	F1	JGA	SGA	F1
KLD+Fuse	21.04	67.13	39.74	23.84	62.94	59.80	19.41	54.61	31.03	24.57	51.28	27.24	60.00	75.56	68.78
KLD	19.63	65.49	32.68	24.19	60.91	52.04	16.55	54.71	29.54	18.00	47.04	19.54	60.06	74.73	64.49
Fuse	18.89	65.01	34.20	22.13	57.12	48.89	15.87	54.65	31.91	21.70	47.98	22.35	59.48	73.24	62.33

TransferQA, which combines candidate choice tokens with the input question and context that tend to be truncated due to the lengthy inputs. Furthermore, we maintain context information by fusing the contextualized weights of candidate choices for inference. Experimental results demonstrate our proposed method effectively generalizes from QA to new domains without annotated data for DST. The JGA is somewhat less in the other two domains where multiple-choice slots mostly contain an extensive collection of choices to be efficiently used for context-choice fusion. For example, the average number of choices for categorical slots in the domain ontology of “*restaurant*” is 5, while that of “*attraction*” is 11. It indicates that more choices associated with a slot lead to less efficient incorporation of choice knowledge into the model. Compared to the first three rows in subsection A.1.4 based on leave-one-out training, our model outperformed all of them, indicating it is unnecessary to perform in-domain training.

4.2.5 Discussion

We evaluate two components in the choice-fusion mechanism, i.e. the appreciative choice selection with KLD loss (KLD) and the context-choice fusion ($Fuse$). Table 4.2 summarizes the results of using our proposed techniques, where the slot goal accuracy (**SGA**) is the average accuracy of predicting the value of a slot correctly and F1 is the harmonic mean of the precision and recall to evaluate the performance of slot-value predictions. It is evident that the model adopting both modules essentially outperforms the models with only one module. SGA and F1 drop when not using the context-fused result for the decoder,

indicating that context fusion is important for constraining the generation of slot-values at the slot level. It is noted that JGA drops slightly for all except “train” and “restaurant” domains compared to applying KLD loss only. This suggests that the turn-level generation of dialogue states could potentially be disturbed by re-initializing the decoder, while slot-level accuracy and F1 still outperform. Moreover, it is observed that in all cases metrics drop significantly without KLD loss, which proves that the improved alignment of the prior and posterior distributions of choices will benefit accurate value generation. Empirically, it seems that leveraging ground-truth annotations is better than only fusing choices dependent on the prior context in training because the appreciative choice selection is superior to context-choice fusion on all metrics.

The SGA by domains and slot types in subsection A.1.4, shows that our model outperforms three more challenging domains with more non-categorical slots than categorical ones. Non-categorical slots tend to have a larger vocabulary size of values and thus benefit from joint training using knowledge fusion. Another reason might be owing to more extractive data used in QA training, which makes our model generate extractive values better. As our focus in this paper is on choice selection and fusion, we leave the study on related datasets as future work.

4.3 Dialogue State Tracking Based on Schema-Guided Knowledge Graphs

4.3.1 Motivation and Contribution

LLMs achieved remarkable performance in task-oriented DST by fine-tuning batches of task-oriented dialogues [63, 58, 90, 164]. However, their performance is at the cost of a large number of computational resources as the sizes of LLMs increase rapidly. Recently, prompt-based adaption methods are proposed, which freeze the LLM while only allowing a small number of parameters updated for downstream tasks [106, 107, 108]. Such prompt-based adaption eases the computational cost in fine-tuning large language models per downstream task, and thus improves the scalability of domain adaption. Generally,

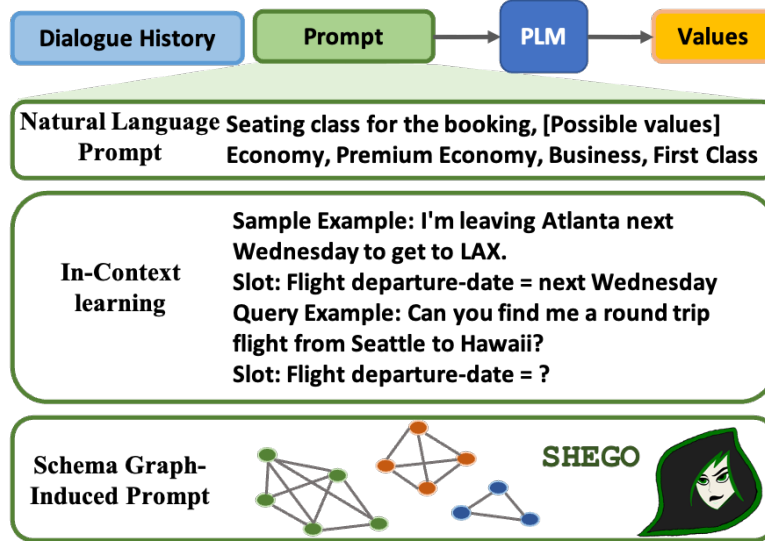


Figure 4.2: Illustration of prompt-based adaptation methods: natural language prompt, in-context learning, and schema graph-induced prompt.

there are two paradigms of prompt-based adaptation methods for task-oriented DST: 1) *Natural Language Prompt* uses schema descriptions and possible values in natural language to draw domain-specific knowledge from the LLM [112, 113, 114]; 2) *In-Context Learning* prepends related task examples to condition on the generated dialogue states [115, 116, 117]. However, the effectiveness of both paradigms highly depends on how well the conditional prompts fit in the input of LLM [108]. Therefore, the prompt can update automatically and model domain-specific schema efficiently for multi-domain DST is demanding.

To address these issues, we propose a **ScHEma Graph-guided prOmpt** (SHEGO) paradigm based on graph neural networks (GNNs) [165, 166] to incorporate the schema for domain-specific prompt learning [36]. All three paradigms are illustrated in Figure 4.2. Specifically, we model the multi-domain schema as a graph of slot tokens with their relations. GNNs are effective feature extractors using structural relations of domain-specific characteristics to condition on generating slot-value pairs and have been proved in previous work [167, 111]. Besides, we leverage several trainable tokens shared by all domains to adapt the pre-trained language model to DST. Then, dialogue history, slot tokens, and trainable shared tokens are combined in a row. The schema graph represents domain- and service-specific

information and relevant structural relations then are aggregated by ASAP pooling [168]. In other words, our graph-structured prompts are embedded through a GNN. In the end, we freeze the entire pre-trained model except for the GNN and embedding layers for dialogue history and the shared soft prompts. Thus the limitation imposed on the parameters can help prevent overfitting to specific domains [108].

Previous work has shown the efficiency and effectiveness of formulating DST in a sequence-to-sequence manner by predicting slot-value pairs all at once [58, 112, 113]. We thus formulate DST as masked span filling, which replaces desired values in the query with mask tokens and generates spans for masks as the output. Such a value-masking formulation mimics the pattern in pre-training of language models which bridges the gap with downstream dialogue state tracking, and has been proven effective in previous work [140, 141, 114]. We use this to control the dialogue state generation with efficiency, as well as to leverage capabilities of natural language understanding in frozen LLM.

With T5-small [155] as the backbone LLM, we evaluate SHEGO on two well-known DST benchmarks: Schema-Guided-Dialogue(SGD) [82] and MultiWOZ 2.1 [159]. We show that our method achieves significant improvement in multi-domain dialogue state tracking on both benchmarks. On the dataset level, SHEGO achieves state-of-the-art performance with a significant reduction of tunable parameters compared to existing models on Schema-Guided Dialogue dataset (SGD). On the domain and service level, SHEGO has consistent gains by a margin on almost all domains, especially in domains with more slots.

Our work conducts a comprehensive study of domain-level representation learning methods for DST and carefully examines their differences and commonalities in terms of performance and characteristics. Furthermore, we propose a parameter-efficient framework for DST that can incorporate existing methods as submodules and automatically learn to activate the appropriate ones for a given domain and context.

Our contribution is as follows: (1) We propose SHEGO, a graph-based prompt learning method that first incorporates slot relations in the schema and learns domain-aware

prompts from domain-specific schema. (2) We conduct a comprehensive study of using GNN as an encoder for prompts for task-oriented DST and examine their differences and commonalities in terms of performance and characteristics. (3) Our experiments on SGD and MultiWOZ 2.1 demonstrate the effectiveness of the schema graph-induced prompts that enable domain adaptation with limited trainable parameters².

4.3.2 Method

We introduce SHEGO, a prompt learning approach based on GNN—*i.e.*, a graph convolutional neural network (GCN)—that learns graph prompt embeddings from the schema. Figure 4.3 shows the architecture of our proposed method. Specifically, the GCN layers encode relations among all slots in the dialogue schema into embeddings, which are then combined with embeddings of shared soft prompt tokens. The LLM, *i.e.* T5-small [155], and other parts of the input are fixed, whose weights are drawn from pre-training. elaborate on DST task formulation (subsubsection 4.3.2), a GNN-based prompt encoder (subsubsection 4.3.2) and prompt training (subsubsection 4.3.2) on the following:

Masked Slot Filling

In dialogue state tracking, the schema defines each domain \mathcal{D}_l and its corresponding slots \mathcal{S}_l in a tuple $\{\mathcal{D}_l : d^{D_l}, \mathcal{S}_l\}$, where $\mathcal{S}_l = \{s_1 : d_1, \dots, s_{n_l} : d_{n_l}\}$ and d are descriptions for domains and slots. The objective of DST is to output $Y = \{(s_1, v_1), \dots, (s_{n_l}, v_{n_l})\}$ as slot-value pairs, given the dialogue history $H = [u_1^{sys}, u_1^{usr}, \dots, u_t^{sys}, u_t^{usr}]$ as the concatenation of the system and user utterances in previous turns, where t is the number of current turns in the dialogue.

We experiment with T5, which is an encoder-decoder model pre-trained to predict corrupted tokens in the input, using sentinel tokens as the hint [141]. To take after the objective in pre-training, we substitute each value in Y with a sentinel token as well. Specifically,

²Code is available at <https://github.com/youlandasu/Graph-Adapter>

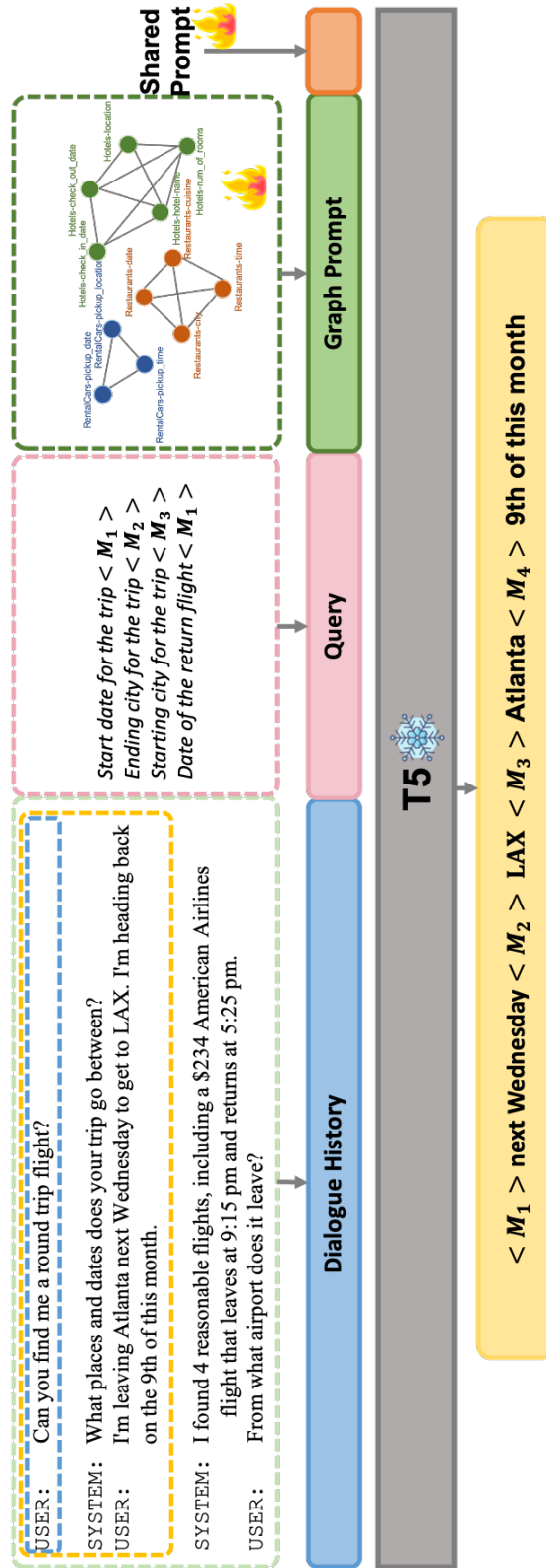


Figure 4.3: The overview of schema graph-induced prompt (SHEGO) architecture for multi-domain DST.

we form a query for each input sequence by combining sentinel tokens of slots with their descriptions into a sequence:

$$Q_l = [q_1, \dots, q_{n_l}], \text{ where } q_j = d_j : \langle \mathcal{M}_j \rangle \quad (4.6)$$

d_j are descriptions of the slot, and $\langle \mathcal{M}_j \rangle$ are distinct sentinel tokens of the j -th slot representing masks. Such a combined sequence is then appended to the dialogue history. DST is thus formulated as imputing v in the output sequence Y' for each domain.

$$Y' = \langle \mathcal{M}_1 \rangle v_1 \langle \mathcal{M}_2 \rangle v_2 \dots \langle \mathcal{M}_{n_l} \rangle v_{n_l} \quad (4.7)$$

To leverage the power of trainable prompt tokens and reduce the computational costs, we append $m = \sum_l |n_l|$ graph prompts G and p shared soft prompts P to the input sequence.

$$G = G_1 G_2 \dots G_m \quad (4.8)$$

$$P = P_1 P_2 \dots P_p \quad (4.9)$$

In the following, we generate structural representations of prompts G by the schema-graph encoder and train embeddings of soft prompt P .

Schema Graph Prompt Encoder

Our schema graph prompt encoder is designed to model slot relations in the dialogue schema and extract their global characteristics. These structural representations are used to condense the dialogue representation and fit in the dialogue input to the frozen pre-trained language model. Below we describe the structure encoder for domain ontology.

We model the slot level relations by building up an undirected graph for the i -th dialogue, *i.e.* $\{G^i, \mathcal{A}_k^i\}$, where G^i are distinct prompts of pre-defined slots in training dialogues and each node represents a particular slot. $\mathcal{A} \in R^{m \times m}$ is the adjacency matrix, m is the

number of domain-slot pairs in the training data. Denoting the set of edges as $E = \{(u, v)\}$ and $u, v \in G^i$, we build an edge from a particular node of slots to another node belonging to the same domain. These edges incorporate slot-level semantics contained in the schema.

Moreover, we transform $G^i \rightarrow \tilde{G}^i$ by masking G^i with active (not none) slots in the i -th dialogue. Specifically,

$$\tilde{G}_j^i = \begin{cases} G_j^i & \text{if } j\text{-th slot active} \\ 0 & \text{otherwise} \end{cases} \quad (4.10)$$

The motivation behind this is that active slots are contextual information for each dialogue turn in training. We hypothesize that this setup improves the diversity of input graph prompts, such that adapts to the task more promptly.

We use a hierarchical graph pooling-based encoder, which includes multiple layers of GCN each followed by adaptive structure-aware pooling (ASAP) layer [168]. The GCN analyzes the input graph to create structurally informed representations of the nodes. ASAP then groups similar nodes and ranks the clusters, forming new nodes and edges from the top clusters. This process generates a structurally informed graph representation and allows us to utilize shared information from ontology in other domains.

In this way, the graph representation is learned and accumulated preserving the structural information. After each pooling step, the graph representation is summarized using the concatenation of the mean and max of the node representations. The summaries are then added and passed through fully connected layers to obtain the final structural representation of the domain ontology.

End-to-End Training

We combine the dialogue history H , dialogue query Q_i^i , graph prompts G , and p shared soft prompts P together as the input to frozen T5. Prompts G and P can be viewed as the flexible context that helps the fixed pre-trained model adapt to DST tasks in specific domains. With the input augmented with trainable graph prompts and shared soft prompts,

Table 4.3: Accuracy on each domain on the SGD test set. AdapterCL[†] is drawn from their best model’s results, and Prompt-Tuning* is drawn from our re-implementation.

Model	Pretrained Model	Avg. JGA
SGD-baseline	BERT-base / 110M	25.1
Seq2Seq-DU	BERT-base / 110M	22.1
AdapterCL [†]	GPT-2 / 1.5B	39.7
Prompt-Tuning*	T5-small / 60M	73.1
SHEGO	T5-small / 60M	76.6 _{±3.4}

we train our model in an end-to-end manner.

Our model maximizes the conditional probability of Y' by training the parameters θ_P and θ_Q of the prompts, while all other parameters are fixed:

$$\mathcal{L}_{\theta_P, \theta_Q} = - \sum_l \sum_i^{|D_k|} \log p_{\theta}(Y' | H^i, Q_l^i, \tilde{G}^i, P^i) \quad (4.11)$$

4.3.3 Experiment

Dataset

We verify the effectiveness of our models on widely used benchmarks for multi-domain task-oriented dialogue state tracking: Schema-Guided-Dialogue(SGD) [82] and MultiWOZ [159]. The SGD dataset provides 38K training examples of 44 services over 19 domains, following pre-processing from [140]. The schema defines each slot and corresponding service, with a brief description of the slot and service. We only consider dialogues of a single service and take $Service_l$ as \mathcal{D}_l in our method as in [141]. To make the comparison fair, we calculate the accuracies of several tasks from the same domain and present our presented results by domain.

We evaluate models on MultiWOZ 2.1 [159] and follow dataset setups in [58] that ex-

Table 4.4: Accuracy of our model and baselines on SGD dataset by domains. Our model outperforms baselines on joint and slot goal accuracies. AdapterCL[†] is drawn from their best model’s results, and Prompt-Tuning* is drawn from our re-implementation.

Domain	SGD-baseline		Seq2Seq-DU		AdapterCL [†]		Prompt-Tuning*		SHEGO	
Metric	JGA	AGA	JGA	AGA	JGA	AGA	JGA	AGA	JGA	AGA
<i>Travel</i>	41.5	57.2	44.9	-	39.6	58.9	76.2	81.5	80.4 _{±9.2}	88.8 _{±4.5}
<i>Weather</i>	62.0	76.4	57.9	-	72.5	86.0	95.5	97.1	97.0 _{±1.5}	98.1 _{±1.0}
<i>RideSharing</i>	17.0	50.2	67.0	-	61.5	81.3	91.0	97.4	86.3 _{±3.7}	94.7 _{±1.3}
<i>Homes</i>	18.9	72.7	22.8	-	45.4	71.9	81.9	91.0	82.6 _{±3.8}	94.9 _{±1.9}
<i>Hotels</i>	28.9	58.2	34.0	-	31.3	62.5	73.6	91.5	80.4 _{±2.8}	93.3 _{±0.8}
<i>Movies</i>	37.8	68.6	43.9	-	26.9	40.4	65.1	90.9	72.5 _{±3.6}	92.5 _{±2.0}
<i>Services</i>	40.9	72.1	47.7	-	28.6	69.4	74.3	89.6	77.2 _{±3.5}	91.4 _{±1.5}
<i>Buses</i>	9.7	50.9	16.8	-	23.1	56.9	78.9	94.4	81.3 _{±3.2}	94.9 _{±1.0}
<i>Payment</i>	11.5	34.8	7.2	-	58.8	87.7	37.1	65.5	43.4 _{±12.0}	81.4 _{±4.2}
<i>Trains</i>	13.6	63.5	16.8	-	24.4	62.3	60.7	88.7	65.8 _{±10.9}	90.6 _{±2.7}
<i>Music</i>	15.5	39.9	12.3	-	21.4	61.0	62.1	84.2	64.7 _{±2.7}	84.1 _{±3.0}
<i>RentalCars</i>	8.6	48.0	6.25	-	20.5	61.7	64.8	91.6	72.6 _{±3.4}	93.0 _{±1.3}
<i>Restaurants</i>	22.8	55.8	13.0	-	16.9	61.0	67.0	92.3	67.9 _{±3.2}	92.3 _{±1.6}
<i>Events</i>	23.5	57.9	31.8	-	28.6	60.2	77.6	94.2	79.9 _{±2.6}	95.1 _{±0.8}
<i>Flights</i>	23.9	65.9	15.9	-	22.0	52.8	77.6	95.5	80.8 _{±4.0}	96.3 _{±0.8}
<i>Alarm</i>	57.7	1.8	55.7	-	58.8	87.7	87.8	90.5	87.9 _{±1.6}	93.2 _{±2.9}

periments with 5 most frequent domains in the dataset: $\{restaurant, hotel, train, attraction, taxi\}$ and 30 slots. 12K examples and slot-level descriptions are used for training. Similarly, we consider $Domian_l$ in MultiWOZ 2.1 as D_l in our method. Detailed statistics of datasets are presented in section B.1.

Metric

We use joint goal accuracy (JGA) and average goal accuracy (AGA) to evaluate our models and baselines. Joint goal accuracy is the average accuracy of predicting all slot-values for a turn correctly, while average goal accuracy is the average accuracy of predicting the value of a slot correctly for active slots in the ground truth. A slot is called active if its value is NOT none in the ground truth dialogue state of the current turn. We compute JGA and AGA on SGD by domains and MultiWOZ datasets, reporting the mean and the standard deviation of 3 random runs. We also report Avg. JGA defined as the average of JGA across

all L domains:

$$\text{Avg. JGA} = \frac{1}{L} \sum_{l=1}^L \text{JGA}_{\mathcal{D}_l} \quad (4.12)$$

Implementation details We implement our approach based on T5-small (60M parameters) [155] pre-trained language model whose hidden dimension is 512. Parameters inside of T5-small are fixed, with only GNN layers and embeddings for prompts trainable at a batch size of 16 using AdamW [169] optimizer. The initial learning rate is set to 0.01 with weight decay $5e - 4$ for GNN layers and 0.5 for T5. We use greedy decoding for the maximum length of 100 for all models. Early stopping is adopted if validation performance does not improve for 5 consecutive epochs. The number of prompts for the schema graph is equal to that of domain-slot pairs in all the schema. Tokens representing domain-slot pairs are augmented to the model’s vocabulary and their embeddings are initialized with normal distribution. The hidden dimension of the ASAP layer is 256 while its output layer has a size of 512. We use embeddings that enumerate 100 additional tokens in the vocabulary as the additional tunable prompt initialization with the embedding size 512.

Baseline

We evaluate the performance of SHEGO for multi-domain DST compared with the following baseline models. **SGD-baseline** [82] fine-tunes BERT to predict values for each slot. **Seq2Seq-DU** [164] employs BERT in the encoding of utterances and schema descriptions respectively and generates pointers in decoding. **TRADE** [58] adopts slot gate and copy mechanism to track slot values mentioned in the dialogue history **TripPy** [90] enhances DST with a triple copy mechanism exploiting system inform memory as an extra input. **TripPy-R** [170] combines copy mechanism in TripPy with a unified encoder and attention mechanism to improve slot matching. **T5DST** [112] proposes schema-driven prompting to improve DST in low-source scenarios. **MoNET** [171] leverages contrastive context matching to update and correct slot values. **PPTOD** [172] presents a unified plug-and-play model for DST with task names as prompts. **SeKnow-PLM** [173] improves dialogue modeling

Table 4.5: Joint and slot goal accuracy on each domain on MultiWOZ2.1.

Model	Pretrained Model / #Param.	<i>JGA</i>	<i>AGA</i>
SGD-baseline	BERT-base / 110M	43.4	-
TRADE	-	48.6	96.9
Seq2Seq-DU	BERT-base / 110M	56.1	91.1
TripPy	BERT-base / 110M	55.3	-
T5DST	T5-base / 220M	56.7	-
MoNET	BERT-base / 110M	57.7	97.7
PPTOD	T5-large/770M	57.5	-
SeKnow-PLM	GPT-2 / 1.5B	58.5	-
Trippy-R	RoBERTa-base / 120M	56.0	-
DiSTRICT	T5-small / 60M	56.1	-
DiCoS-DST	ALBERT-large / 18M	61.0	98.1
SHEGO	T5-small / 60M	59.0 \pm 0.6	95.6 \pm 0.1

grounded on semi-structured knowledge. **DiSTRICT** [117] utilizes retrieved in-context examples to fine-tune the language model. **DiCoS-DST** [135] explicitly models slot dependency to select relevant contents dynamically. **AdapterCL** [140] learns an adapter to avoid catastrophic forgetting in continual learning. **Prompt Tuning** for DST [141] first proposes to use prompt tuning for DST focusing on using prompts to transfer knowledge in continual learning,

4.3.4 Results and Analyses

We show the effectiveness of our models on Schema-Guided-Dialogue(SGD) (subsubsection 4.3.4) and MultiWOZ (subsubsection 4.3.4) compared with baselines, respectively. And we show the model performance when the ASAP layers are initialized with different architectures in subsubsection 4.3.4. In subsubsection 4.3.4, we investigate our proposed approaches with a comprehensive ablation study.

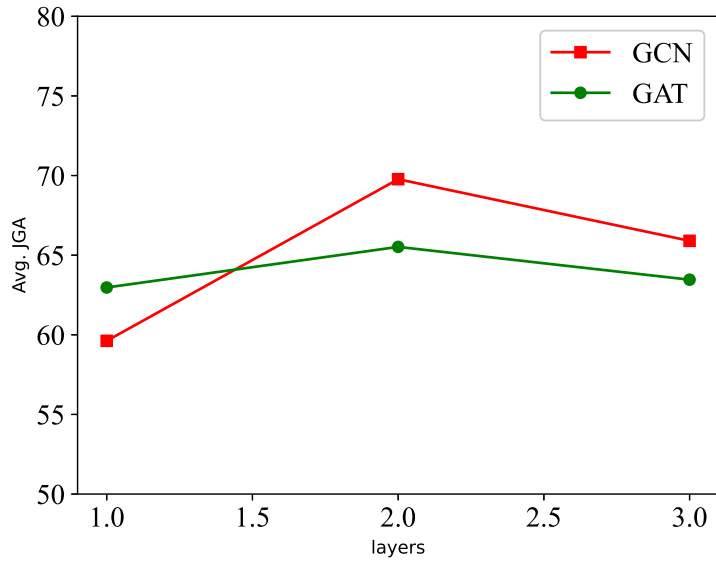
Results on SGD

Table 4.3 shows the evaluation results of using T5-small on the SGD test set. The results are averaged over three random seeds. Our proposed approaches with frozen T5-small achieved the state-of-the-art JGA by 3.5%, even outperforming GPT-2 with 1.2B parameters. We further verify that our models achieve consistent gains over almost all domains on SGD, as shown in Table 4.4.

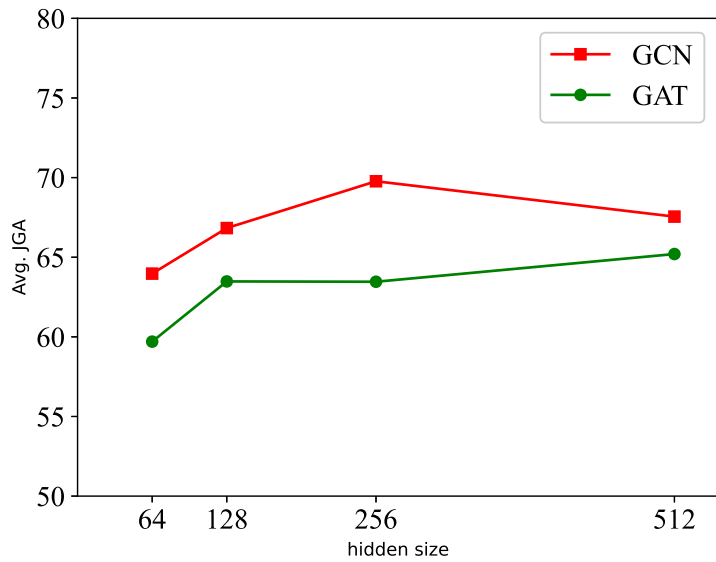
Our observations are as follows: 1) With notably less trainable parameters, SHEGO achieves a prominent and consistent margin over SGD-baseline and Seq2Seq-DU. This indicates that SHEGO can obtain generalized models for DST with higher parameter efficiency. 2) SHEGO outperforms other parameter efficient approaches, *i.e.* AdapterCL and Prompt-Tuning, suggesting that inducing schema into models effectively improves the performance of learning trainable prompt parameters. 3) As shown in Table 4.4, SHEGO outperforms sixteen out of eighteen domains compared to all baselines under the full-data setting, except for the “*RideSharing*” domain (-4.7 from Prompt-Tuning) and “*Payment*” domain. We consider that both have relatively small training size and their slots are different from those of other domains, thus comprising the benefit of scheme graph inducement and multi-domain joint training. 4) Our method has consistent gains by a large margin, especially in domains with a large number of slot types, *e.g.* “*Movies*”, “*Hotels*”, “*Flights*”, suggesting that inducing the schema graph to improve the performance for domains with sufficient slot relations.

Results on MultiWOZ

Table 4.5 shows joint goal accuracy on the MultiWOZ 2.1 test set. Our model achieves a competitive JGA of 59.0 on MultiWOZ 2.1 with about 10M tunable parameters. Compared with fine-tuning models, we learn 2 ~ 15 times fewer parameters in SHEGO. This suggests that, instead of fine-tuning, SHEGO still improves the performance of multi-domain DST with induced schema graphs and shared prompt embeddings. Our model is still some-



(a) Analysis of the number of pooling layers for GCN and GAT. The hidden size for GNNs is set to 256.



(b) Analysis of hidden size for GCN and GAT. The number of layers is set to two.

Figure 4.4: Comparison of architectures of graph neural networks. Results are shown in Average JGA over the first 15 services in SGD as listed in Table B.3.

Table 4.6: Ablation results on the first 15 services of SGD listed in Table B.3, section B.2. All numbers are reported in average joint goal accuracy(JGA) (%) over 15 tasks. *w/o Active* means the graph module encodes the same schema information in training. *w/o SlotConnect* indicates the model is trained without graph modules while only adding the same amount of prompt tokens as graph modules.

No.	Model	Applied Strategy			Avg. JGA
		GNN	Graph Prompts	Active Slots	
1	<i>w/o Active&GP</i>	✓			63.5
2	<i>w/o GP</i>	✓		✓	64.6
3	<i>w/o Active</i>	✓	✓		67.5
4	<i>w/o SlotConnect</i>		✓	✓	66.7
5	Our Model	✓	✓	✓	69.8

what worse than prior DiCoS-DST. This is likely because the DiCoS-DST model utilized ground truth for modeling relations of the slots in the current turn with the last updated turn. SHEGO also obtains +2.9 on DiSTRICT with the same backbone models, showing that trainable parameters enable greater scalability for our proposed model compared to the reference in-context learning approach. Prompt Tuning for DST is the closest work to ours, while we use GNN to model structural relations in the schema and promote the flexibility of prompts.

Analysis of Schema Graph Architectures

In Figure 4.4, we compare using GAT and GCN in the schema graph and the number of layers for pooling, where the hidden dimension is 256. We evaluate the first 15 services of SGD as listed in Table B.3 and report the average joint goal accuracy over these services. The results show that GCN with two layers outperforms other GNN settings, which is the setup we adopt in our model. We further study the impact of hidden size for GCN and GAT varying in {64, 128, 256, 512}. As illustrated in Figure 4.4, the hidden size of 256 exceeds others performing the same tasks as above. Based on this, we remark that there is a trade-off between the number of parameters trained from scratch as prompts and the size

of the training set. The results also suggest that even though GNN with more hidden layers can preserve more generalized schema information, generalized schema information might not be fit for prompting from the frozen pre-trained model.

Ablation of Schema Graph Prompt

To investigate the effectiveness of the embeddings encoded by the schema graph, we conduct an ablation study on our proposed schema graph with variable settings. Our experiments consist of the following: 1) w/o Active&GP turns each graph prompt (GP) into the same token from the model’s vocabulary, instead of using graph prompts and masking by the active slots as in our original model. 2) w/o GP represents substituting each graph prompt with the same token from the model’s vocabulary, and adopts active slot masking for each input example. 3) w/o Active encodes graph prompts with GNN but is not masked by active slots. 4) w/o SlotConnect adds shared soft prompts at the same length as graph prompt length and is trained the same with the shared prompts. It is similar to learning parameters of graph prompts in 3) without graph neural network.

Results are presented in Table 4.6. First, we observe that formulating input slot nodes as trainable prompts brings about improvement compared to using active slots indices only. Comparing 5) to 2) and 3) to 1), tunable graph prompts enhance Avg. JGA by 4.8 and 4.0. It gives additional trainable parameters to the graph neural network to learn the characteristics of each domain-specific feature. Second, incorporating active slots in each input example is beneficial to generate masked spans correctly. Comparing 5) to 3) and 2) to 1), masking with active slots help improve JGA by 2.3 and 1.1, respectively. But the gain from indices of active slots is weaker than that from the prompt token formulation. Third. It is evident to show the effectiveness of our schema graph using a graph neural network comparing 5) with 4). Prompt tokens improve by 3.1 on the accuracy of learning slot connections within the graph neural network. Moreover, prompt tokens cannot benefit from active slots in our empirical experiments, which might be because they make input features too sparse to train

an embedder without graph pooling.

4.4 Conclusion

This chapter has presented two innovative approaches that significantly advance the field of dialogue state tracking through enhanced domain-specific learning. Our research demonstrates the substantial benefits of incorporating both domain transfer with slot-values and schema graph-guided prompts into state tracking systems.

The traditional approaches to dialogue state tracking, while functional, have shown limitations in their ability to scale across domains and handle new scenarios without extensive domain-specific annotations. Our investigation has revealed that these limitations stem from fundamental assumptions about how dialogue states should be tracked and represented, particularly the heavy reliance on supervised learning with domain-specific data.

The Choice-Fusion DST approach represents a significant advancement in addressing these limitations. By explicitly modeling candidate choices and leveraging the inherent structure of domain ontologies, this approach demonstrates superior performance in zero-shot scenarios while maintaining interpretability. The fusion mechanism’s ability to combine information from multiple sources - dialogue history, domain schema, candidate choices, and previous states - provides a more robust foundation for state tracking than traditional approaches.

The proposed methods not only advance the state-of-the-art in DST but also provide practical solutions for building more robust and adaptable dialogue systems. Graph-based representations offer a promising direction for handling the complexity of multi-domain dialogues. The schema-guided prompt approach presents a scalable solution for domain adaptation, addressing one of the key challenges in deploying dialogue systems in real-world applications.

These advances represent important steps toward more robust and adaptable task-oriented dialogue systems, capable of handling increasingly complex and diverse user interactions.

As we continue to build upon these foundations, we move closer to dialogue systems that can truly understand and respond to user needs across a wide range of domains and contexts.

CHAPTER 5

INCORPORATING DIALOGUE-LEVEL KNOWLEDGE FOR TASK-ORIENTED DIALOGUE SYSTEM

5.1 Introduction

As discussed earlier, we define domain-specific and dialogue-level knowledge that drive dialogue flow in TOD systems. The first consists of domain-specific rules, which support inference about goal-directed activities by mapping user intents to structured representations. The second, and more fundamental, consists of domain-independent rules related to system and user dialogue acts. Unlike domain-specific knowledge, dialogue-level knowledge is independent of domain predicates and enables the application of general conversational structures across domains while dynamically interacting with domain-specific content.

This chapter explores advanced methodologies for incorporating dialogue-level knowledge into TOD systems, with a focus on two key techniques: Act-Aware Slot-Value Prediction and Soft Mixture-of-Experts. These methods address two fundamental challenges in dialogue systems, improving DST accuracy and enabling scalable multi-domain adaptation. Through an in-depth analysis, we demonstrate how structured dialogue knowledge can significantly enhance the efficiency and robustness of TOD systems.

The main objectives of this chapter are threefold. First, we examine how dialogue acts can be systematically integrated into state tracking to improve accuracy and adaptability. Second, we explore scalable architectures that enable efficient handling of multiple tasks and domains through knowledge-driven learning frameworks. Finally, we present empirical evaluations and case studies that highlight the practical impact of these approaches in real-world dialogue applications.

Table 5.1: An example of cross-domain dialogue with dialogue state representations and system dialogue acts in MultiWOZ 2.1

USER: Hi, I am looking for a train that is going to cambridge and arriving there by 20:45, is there anything like that?
Dialogue States:
TRAIN: *destination=cambridge, arriveby=20:45*

SYSTEM: Where will you be departing from?
USER: I am departing from Birmingham New Street.
Dialogue States:
TRAIN: *destination=cambridge, arriveby=20:45, departure=birmingham new street*
Dialogue Acts: Inform, Request

SYSTEM: Can you confirm your desired travel day?
USER: I would like to leave on Wednesday.
Dialogue States:
TRAIN: *destination=cambridge, arriveby=20:45, departure=birmingham new street, day=wednesday*
Dialogue Acts: Request

SYSTEM: I have booked your train tickets, and your reference number is a9nhso9y.
USER: Thanks so much. I would also need a place to stay. I am looking for something with 4 stars and has free WiFi.
Dialogue States:
TRAIN: *destination=cambridge, arriveby=20:45, departure=birmingham new street, day=wednesday*
HOTEL: *stars=4, intenet=yes, type=hotel*
Dialogue Acts: OfferBooked

5.2 Incorporating Knowledge from Dialogue Acts

5.2.1 Motivation

In a human dialogue, speech acts are illocutionary actions contained in utterances that change dialogue states [174], for example, a promising, request, affirmation, complaint, or piece of information. The speech act theory has been widely applied in computational communication theories, where responses to a speaker's utterances are guided by the hearer's recognition of the intentions underlying the utterances [52]. Despite syntactic and semantic analysis, speech acts involve higher-level analysis requiring reference to external infor-

mation, i.e., discourse context and the speaker’s beliefs and demands. For example, the utterance *It’s raining outside.* has a syntactic form of a declarative and a literal semantic of the type of weather. However, the utterance having this literal reading in a different context could function as a request for an umbrella or closing a window. Similarly, those representing the illocutions of utterances in a human-machine dialogue are known as **dialogue acts**. For example, the dialogue act “*Request*” in an utterance typically represents an illocutionary action for soliciting information about the task. Inspired by the logical relation of speech acts and utterances in human dialogues, we explore similar logical relations between dialogue acts and utterances in human-machine dialogues. Representations of dialogue states are dependent on not only domain-specific rules and but also domain-independent rules. Table Table 5.1 shows an example of state representations within a task-oriented service dialogue between a user and a system for service across train and hotel domains. Generally speaking, dialogue acts are defined in dialogue-level ontology and serve the functions of conducting particular tasks, having the potential to guide user utterances and enhance the performance of DST as auxiliary inputs. Table 5.2 shows system dialogue acts in MultiWOZ 2.1, consisting of 13 acts serving illocutionary functions in dialogues for reservation service across 7 domains.

In previous work, speech acts are taken as noisy observations of dialogue acts to update dialogue states with generative methods [56, 57] which represent dialogue states at each turn by modeling the joint probabilities in belief space. Such methods provide more reasoning for dialogue systems, while they are not robust to outliers and limit the flexibilities of dialogue models. More recent works represent dialogue states as a set of slot-value pairs [175, 176], where discriminative models have proved their capabilities in tracking dialogue states, by modeling DST as multi-task classification [58, 59, 60, 61] or question-answering problems [62, 63, 64, 65]. Recent discriminative DST models estimate user goals directly from the dialogue context, ignoring the NLU module and dialogue acts. However, using dialogue acts not only helps reason how dialogue states are predicted but

Table 5.2: System Dialogue Acts in MultiWOZ 2.1, consisting of 13 acts serving illocutionary functions in dialogues for reservation service across 7 domains

System Act	Dialogue Utterance	Frequency Rate
Inform	The address is 64 Cherry Hinton Road.	36.03%
Request	What day would you like your booking for?	16.41%
reqmore	Is there anything else I can assist you with?	12.65%
bye	Thanks and enjoy your time!	8.37%
Book	Your booking was successful. The table will be reserved for 15 minutes.	4.83%
welcome	You're welcome. Hope you have a great trip.	4.40%
Recommend	Then I recommend the A and B guest houses.	4.09%
OfferBook	The TR3877 leaves at 12:09 on Friday. Would you like me to book it for you?	2.79%
NoOffer	I am sorry, we do not have any Greek restaurants any where in town.	2.73%
Select	I found 2 expensive hotels, 1 in the centre and 1 in the south of town. Do you have a preference?	2.53%
OfferBooked	I was able to book you for 4 tickets on train TR7278. The total fee is 51.2 GBP payable at the station.	2.12%
greet	I can help you with that.	1.86%
NoBook	That booking is unsuccessful.	1.21%

also improves the compatibility of the DST model to existing pipeline dialogue systems and scalability to new domains.

Combining recent advances in discriminative dialogue modeling and the speech act theory, we propose an **Act-Aware Dialogue State Tracker (ADST)** [37] to predict slot-value pairs for tracking dialogue states with reasoning and accuracy, incorporating dialogue acts in discriminative DST models. We utilize system dialogue acts because they are easier to acquire and are closely relevant to dialogue system design. Furthermore, we exploit advances in reading comprehension (RC) to extend our DST models for task-oriented dialogues with a more free-form domain ontology. Inspired by the RC-based work on DST [63, 62], we formulate the DST problem as predicting values by querying with two types of slots: pre-defined-value-set (categorical) slots and open-vocabulary (non-

categorical) slots. Figure 5.1 shows the overall architecture of our proposed models¹.

5.2.2 Model

Encoding. Denoting the i -th word in C_t as c_i , we combine word embedding, role embedding, and binary exact match features for c_i as the input to the encoder. Then we use a bidirectional GRU [177] to encode it into X^e , and let the dimension of the encoder’s hidden states be the same as that of the input. The input dialogue act embeddings W^{act} are concatenated word embeddings of system dialogue acts in previous turns. We construct similar word embeddings q^d, q^s for each domain and slot, respectively. As for categorical slots, we sum q^d and q^s with each value embedding w^v in the value set in addition to “none”, “dont_care”, and stack them together to construct option embeddings $F_{d,s}^e$.

Attention on Dialogue Context. We implement an attention model similar to [178] that computes attention weights between dialogue context and slots. Assuming R, S are two matrices with the same columns h , the attention function is defined as:

$$Attention_k(R, S)_j = softmax_i([R_i; S_j; R_i \circ S_j] \cdot k), \quad (5.1)$$

where $k \in \mathbb{R}^{3h}$ is a trainable vector, \circ is element-wise multiplication, $[;]$ is vector concatenation across column. According to the above definition, the attention weights is computed as $\alpha_{d,s}^{k_1} = Attention_{k_1}(X^e, q^d + q^s)$ and $\alpha_{d,s}^{k_1} \in \mathbb{R}^{|C_t|}$. Then the attended slot vector over dialogue context is $Q_{d,s}^c = (X^e)^T \cdot \alpha_{d,s}^{k_1}$.

Attention over Dialogue Acts. In order to fuse information from system dialogue acts, we compute an attended slot vector over dialogue acts implementing an attention model following Equation 5.1. The attention weight of a querying slot attending to acts is given as $\alpha_{d,s}^{k_2} = Attention_{k_2}(W^{act}, Q_{d,s}^c)_{\{d,s\}} \in \mathbb{R}^{|A_t|}$, where W^{act} is the word embedding of the dialogue acts and $|A_t|$ is the total number of system dialogue acts in previous turns. We then obtain a slot vector attended by system acts as $Q_{d,s}^a = (W^{act})^T \cdot \alpha_{d,s}^{k_2}$. Then $Q_{d,s}^a$ is

¹Code is available at <https://github.com/youlandasu/ACT-AWARE-DST>

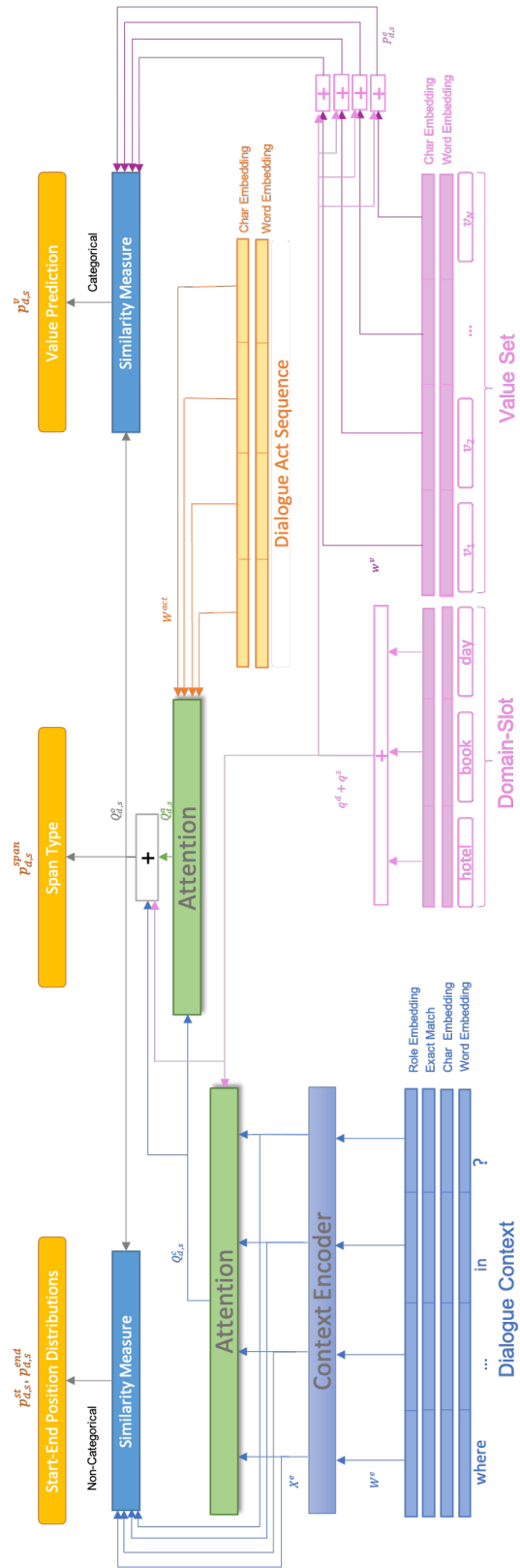


Figure 5.1: Act-Aware Model Architecture for Dialogue State Tracking.

combined with $Q_{d,s}^c$ and the original domain and slot embedding

Value Classification for Categorical Slots. Inspired by [179], we compute probability of a value by calculating the bi-linear similarity between possible options $P_{d,s}^e$ and a final slot representations $Q_{d,s}^o$:

$$p_{d,s}^v = \text{softmax}(P_{d,s}^e \Theta^v Q_{d,s}^o), \quad (5.2)$$

where Θ^v is a trainable weight matrix. We calculate the Cross-Entropy loss L_v between $p_{d,s}^v$ and true value for each categorical slot.

Span Prediction for Non-Categorical Slots. For non-categorical slots, we first decide the type of span from one of the following options: a span can be extracted from the dialogue context, “*dont_care*”, or “*none*”. The probability of the span type is calculated by: $p_{d,s}^{span} = \text{softmax}(FFN_{type}(Q_{d,s}^o))$, where FFN_{type} represents a feed-forward neural network with output dimension of 3. Then we predict the probability distribution of start and end positions in the dialogue context with the following similarity functions:

$$\begin{aligned} p_{d,s}^{st} &= \text{softmax}(FFN_{c_1}(X^e) \Theta^s Q_{d,s}^o), \\ p_{d,s}^{end} &= \text{softmax}(FFN_{c_2}(X^e) \Theta^e Q_{d,s}^o), \end{aligned} \quad (5.3)$$

where FFN_{c_1}, FFN_{c_2} are one-layer feed-forward networks with output dimensions of w , and Θ^s, Θ^e are two trainable weight metrics for predicting the *start* and the *end* position, respectively. Denoting L_{type} as the Cross-Entropy loss of the label type and L_s as the Cross-Entropy loss of start and end positions, we jointly optimize $L = L_v + L_{type} + L_s$.

5.2.3 Results and Analysis

Table 5.3 lists the experimental results on MultiWOZ 2.1[159] test set, where the joint goal accuracy is the average accuracy of predicting all slot values for a turn correctly, while the slot goal accuracy is the average accuracy of predicting the value of a slot correctly. MultiWOZ 2.1 dataset consists of over 10,000 multi-turn dialogues across 7 domains. There are

Table 5.3: Joint and Slot Goal Accuracy on MultiWOZ 2.1

Model	Joint Goal Accuracy	Slot Goal Accuracy
w/o Non-Categorical Slots:		
DS-DST picklist [63]	53.30	-
DSTQA w/o span [62]	51.44	97.24
SOM-DST [180]	53.01	-
CHAN [181]	58.55	98.14
ADST (Ours) all categorical	56.70	97.71
w/ Non-Categorical Slots:		
TRADE [58]	48.62	96.92
STARC [65]	49.48	-
DS-DST [63]	51.21	-
DSTQA w/ span [62]	51.36	97.22
ADST (Ours) all non-categorical	50.17	97.12
ADST (Ours) hybrid	56.12	97.62

13 dialogue acts and 30 (*domain, slot*) combinations with hundreds of possible values in the dataset. We split the dataset into training, development, and test set the same as the original setting, and only use the 5 most frequent domains in the dataset: {*restaurant, hotel, train, attraction, taxi*}. We compare our models with: (1) DS-DST [63] (2) DSTQA [62] (3) CHAN [181] (4) TRADE [61] that generates dialogue states with slot gate and copy mechanism without pre-defined domain ontology, (5) STARC [65] and (6) SOM-DST [180].

We first train a model taking all slots as categorical, comparing it with the categorical-only models: DS-DST picklist [63], DSTQA [62] w/o span, and CHAN [181]. We achieve 56.70% joint goal accuracy and 97.71% slot goal accuracy on categorical-only slot-value predictions, which is close to the state-of-the-art joint and slot goal accuracy on the MultiWOZ 2.1 dataset. It’s important to note that our model is more lightweight and can be scaled to non-categorical slots while the state-of-the-art model cannot. In our hybrid model, we take all number- or time-related slots as non-categorical, whereas other slots as categorical, and train all slots jointly. We compare our results with those of STARC [65], DS-DST [63], and DSTQA [62] with span. We obtain outperformed results of 56.12% and 97.62% on joint and slot goal accuracy, due to the advantages of exploiting RC approaches and using system dialogue acts as auxiliary inputs.

Table 5.4: Ablation Study on MultiWOZ 2.1 Dev Set

Model	Dev Joint	Dev Slot
ADST (Ours) All Categorical	56.89	97.77
- w/o Dialogue Acts	53.95(-2.94)	97.53(-0.24)
ADST (Ours) All Non-Categorical	45.50	96.74
- w/o Dialogue Acts	44.48(-1.02)	96.65(-0.09)

Ablation Study. Table 5.4 shows the ablation study results evaluating our models on either attending previous system dialogue acts to slots or not. The ablation indicates that our models take advantage of input system dialogue acts for predicting slot-value pairs. We also observe that incorporating system dialogue acts into the representations of slots improves the performance of predicting values from the value set by 5% on joint slot accuracy, but has less impact on span predictions of non-categorical slots.

Visualization of Attention Weights. To investigate how dialogue acts impact slot-value predictions, we feed a simulated system dialogue act sequence into our trained categorical-only model and visualize the attention weights between dialogue acts and individual slots in Figure 5.2. Results show that additional attention weights brought by including system dialogue acts suggest correlations between slots and dialogue acts, such that brings about performance improvement on DST.

5.3 Incorporating Knowledge from Task-Oriented Dialogue Pipeline

In today’s world, task-oriented dialogue systems have greatly benefited from LLMs. However, their task-solving performance is constrained by the inherent capacities of LLMs, and scaling these models is expensive and complex as the model size becomes larger. To address these issues, we propose an **Soft Mixture-of-Expert Task-Oriented Dialogue (SMETOD)** [38] system which scales the model capacity for diverse outputs of dialogue systems with significantly less training and inference cost. Specifically, we leverage Soft MoE [182] to improve model capacity and leverage the effectiveness and performance of

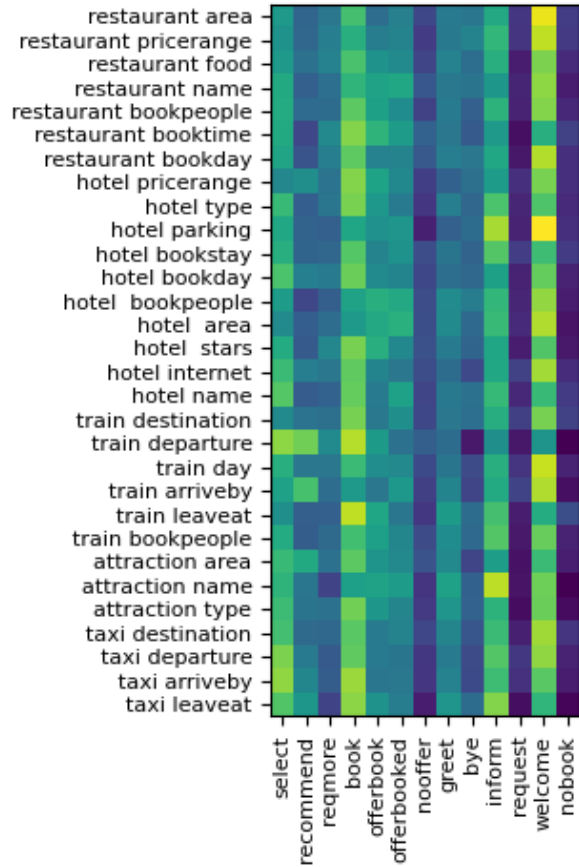


Figure 5.2: Visualization of attention weights on the dialogue act attention layer of our model for categorical slots.

considerably larger models with significantly lower computational costs. We present a task-oriented dialogue system as a multi-module end-to-end text generation to bridge the gap between traditional pipeline-based and end-to-end response generation systems, and optimize NLU, DST, and NLG, respectively, as in [142, 183]. We formulate NLU, DST, and NLG as the text generation problems, which take dialogue history sequence as model input and generate spans as the output. In the cases of NLG, we predict the DST output to obtain the database (DB) state, which becomes incorporated into its input. With T5-small [155] and T5-base [155] as the backbone PLM, we evaluate our method on MultiWOZ [159, 184] and NLU [185, 186, 187] datasets. We show that our method achieves significant improvement in multi-domain DST on MultiWOZ 2.1 and NLG on both benchmarks.

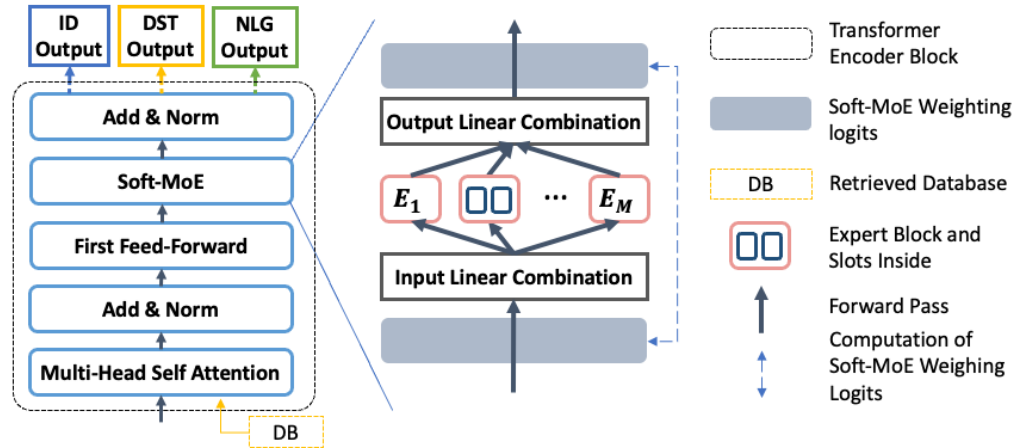


Figure 5.3: Architecture of the **SMETOD** as in Transformer encoders. The result from the DB derived from the output of DST is used for NLG inference. All of the expert layers share the same architecture. The input is ensembled by experts in the Soft-MoE layer for improving model capacity without the cost of efficiency. The model is fine-tuned by maximizing the likelihood of predicting the next token for NLU, DST, and NLG outputs.

Our contribution is as follows²:

- We propose SMETOD, a task-oriented dialogue system for diverse outputs, which first leverages Soft-MoE in text generation and dialogue systems to improve model capacity with efficiency.
- Experimental results demonstrate the effectiveness of our model by improving the performance of NLU and DST on all evaluation benchmarks and achieving comparable performance for NLG.
- Our study of time efficiency and the architect of Soft-MoE proves the significant improvement of efficiency as model complexity continues to grow, promoting future study on dialogue system design with efficiency.

5.3.1 Preliminaries

Soft Mixture-of-Experts

²Code is available at <https://github.com/youlandasu/SoftMoE-TOD.git>.

Table 5.5: Summary of MultiWOZ2.1 dataset [159]. *#non-cat* and *#cat* are the total numbers of non-categorical and categorical slots. Avg. Turns are the average number of turns per dialogue, and Avg. Active are the average number of active (NOT none) slots of each turn.

Split	<i>#Domains</i>	<i>#Dials</i>	Avg. Turns	Avg. Active	<i>#Cat</i>	<i>#Non-cat</i>
<i>Train</i>	5	8437	13.5	2.7	18	12
<i>Dev</i>	5	1000	14.7	2.8	18	12
<i>Test</i>	5	1000	14.7	2.9	18	12

Mixture-of-Experts (MoE)-based models have shown advantages in scaling model capacity without large increases in training or inference costs. There has been work on scaling sparsely activated MoE architectures. In the context of modern deep learning architectures, it was firstly found effective by [188] by stacking MoE between LSTM [189] and resulted in the state-of-the-art in language modeling and machine translation. [190] introduced MoE Transformer where MoE layers are a substitute for the FFN layers.

We adopt Soft-MoE [182], which scales model capacity without the loss of fine-tuning efficiency and is fully differentiable and balanced compared to conventional efficient MoEs [145, 146, 147, 148, 182]. Specifically, it performs a soft assignment on experts to each input token, achieving similar training costs and much lower inference costs at a larger model capacity. We use $f(\cdot; \theta)$ to denote a mapping f associated with the parameter θ from the input sample to an output space. $\sigma(\cdot)$ is the Softmax function. We denote $\{f(\cdot; \theta_i)\}_{i=1}^m$ as m **experts** with identical architectures; their weights $\theta_1, \dots, \theta_m$ applied to individual tokens. Each expert has p **slots**, each of which is a weighted average of input. Slots in the same expert apply the same weights. Given input and output tokens $\mathbf{x} = \{x_1, \dots, x_l\}$ and $\mathbf{y} = \{y_1, \dots, y_l\}$ at the length l . Each expert will process p slots with parameters denoted as $\Psi = \{\psi^{(1)}, \dots, \psi^{(m \times p)}\}$. The input of experts, $\tilde{\mathbf{x}}$, is defined as the result of convex combinations of input tokens.

$$\tilde{\mathbf{x}}_j = (\sigma(\mathbf{x}\psi^{(j)}))^T \mathbf{x} \quad (5.4)$$

where j is the index of the slot in experts and $j \in [1, \dots, m \times p]$. The corresponding expert function is applied on each slot to obtain the output slots:

$$\tilde{\mathbf{y}}_j = f(\tilde{\mathbf{x}}_j; \theta_{\lfloor j/p \rfloor}) \quad (5.5)$$

Given $\tilde{\mathbf{y}} = \{\tilde{\mathbf{y}}_j\}_{j=1}^{m \times p}$, the output of Soft-MoE layer, y_i , is computed as a convex combination of all $(m \times p)$ output slots over the expert dimension (i.e. the rows of $\mathbf{x}\Psi$):

$$y_i = \sigma(x_i \Psi) \tilde{\mathbf{y}} \quad (5.6)$$

5.3.2 Method

We introduce **SMETOD**, a multi-objective dialogue system for NLU, DST, and NLG in task-oriented dialogues, scaling model capacities while maintaining computational efficiency with Soft MoE [182]. The overall architecture is illustrated in Figure 5.3.

Problem Formulation

We define the dialogue history $h = [u_1^{sys}, u_1^{usr}, \dots, u_t^{sys}, u_t^{usr}]$ as the concatenation of the system and user utterances in previous turns, where t is the number of current turns in the dialogue. h has all the dialogue history without the last system utterance, denoted as r . NLU takes h as input and is trained to generate the user intent label text denoted as I , which is an intent or the API-name. The objective of DST is to output user goals, the tasks or purposes that the user wants to accomplish through the dialogue. user goals are typically represented as a set of pre-defined slot-value pairs that consist of the required information to query the dialogue system, i.e. $y_{API} = \{(s_1, v_1), \dots, (s_n, v_n)\}$, where n is the number of slot-value pairs. Finally, NLG will generate S with the previous output: $h + y_{DB} \rightarrow r$, where y_{DB} is the items in the database retrieved by y_{API} . Given a pair of training examples (x', y') , we elaborate x' and y' corresponding to different modules of the dialogue system

in the following Table 5.6.

	x'	y'
NLU	h	I
DST	h	y_{API}
NLG	$h + y_{DB}$	r

Table 5.6: Representations of input and output in task-oriented dialogue system.

Soft Mixture-of-Expert Layer

We implement the Soft-MoE layer to replace the second Feed-Forward Layer in each Transformer [191] Encoder block, as illustrated in Figure 5.3. Mathematically, we denote the output out the first Feed-Forward layer of the k -the encoder is $g(\cdot; \phi_k)$, then $\mathbf{x} = g(x'; \phi_k) \in \mathbb{R}^{l \times d_{ff}}$ in Eq. Equation 5.4, denoting d_{ff} as the dimension between the first and second Feed-Forward layer and d as model’s hidden dimension, and l is the length of tokens. $\psi^{(j)} \in \mathbb{R}^{d_{ff}}$ is d_{ff} -dimensional vector of parameters corresponding to each slot of experts.

The mapping $f(\cdot; \theta_i)$ in Equation 5.5 is simply a linear mapping corresponding to each expert, and p is the slots per expert having the same weights. Therefore, the output of the k -th encoder layer, $y'^{(k)}$, can be represented as

$$y'^{(k)} = f(g(x'; \phi_k); \Theta_k, \Psi_k) \tag{5.7}$$

For fine-tuning, we replicate the pre-trained weights from the second Feed-Forward layer of encoders and assign them to each expert, leveraging the contextual learning abilities inherent in pre-trained models.

Training Objectives

We optimize the generation outputs of NLU, DST, NLG, respectively, following [142]. Given a pair of training samples as (x', y') , the loss function is defined to maximize the

log-likelihood of the token to predict given the current context:

$$\mathcal{L}_{\{NLU,DST,NLG\}} = -\frac{1}{l} \sum_{q=1}^l \log P(y'_q | y'_{<q}; x') \quad (5.8)$$

5.3.3 Experiment

Data

We evaluate our models for NLU on Banking77 [185], CLINC150 [186], and HWU64 [187]; DST and NLG are evaluated on the task-oriented dialogue benchmarks MultiWOZ 2.1 [159] and MultiWOZ 2.2 [184]. Banking77 contains 13,083 customer service queries labeled with 77 distinct intents for distinguishing between intents among queries related to similar tasks. CLINC150, consists of a comprehensive dataset comprising 23,700 examples, annotated with 150 intents across 10 distinct domains. HWU64 is collected from the home robot that has 25,716 examples for 64 intents spanning 21 domains.

MultiWOZ 2.1 [159] consists of multi-turn task-oriented dialogues across several domains, where 8,438 dialogues are for training and 1,0000 for dev and test. MultiWOZ 2.2 [184] improves MultiWOZ 2.1 by correcting annotation errors and adding dialogue act annotations. In MultiWOZ, the generation of response is not only related to the dialogue context but also grounded on the database (DB) state. The DB state is automatically retrieved from a pre-defined database using the generated dialogue states. SMETOD adopts a two-step approach during inference [142, 99]. Firstly, it predicts the DST results to access the DB state. Subsequently, it utilizes the retrieved DB state and the current dialogue context to generate the NLG results.

Training & Inference Details

All models are fine-tuned respectively using PPTOD [142], the pre-trained dialogue models based on T5-small (60M parameters) [105] and T5-base (220M parameters) [105], as the

Table 5.7: Accuracy (%) on three intent prediction datasets with full-data experiments. \diamond comes from Casanueva *et al.*(2020). All others are reported as in the original papers. Models with * are classification-based.

Model	Banking77	HWU64	CLINC150
BERT-FIXED \diamond *	87.19	85.77	91.79
CONVBERT-DG +Pre+Multi*	92.99	92.94	97.11
CONVBERT +Pre+Multi*	93.44	92.38	97.11
BERT-TUNED \diamond *	93.66	92.10	96.93
CONVERT \diamond *	93.01	91.24	97.16
USE+CONVERT \diamond *	93.36	92.62	97.16
SPACE-2 \ddagger *	94.77	94.33*	97.80
SPACE-3*	94.94*	94.14	97.89
TOATOD _{small}	92.40	90.42	98.45
TOATOD _{base}	92.17	90.79	98.01
SMETOD _{small}	92.47	90.88	98.12
SMETOD _{base}	93.02	92.56	98.64

backbone. T5-small has 6 encoders and decoders with hidden size $d = 512$ and $d_{ff} = 2048$. While T5-large has 12 encoders and decoders and $d = 768$, $d_{ff} = 3072$. For models’ architecture, we replace the second Feed-Forward layer in all encoder blocks with the illustrated Soft-MoE layers, and copy pre-trained weights to each expert in the Soft-MoE layers. We augment T5 with 8 experts and 2 slots per expert for DST, and 16 experts with 2 slots per expert for NLU and NLG.

We fine-tuned all model parameters on the full-shot training setting. The linear combination weights in Soft-MoE layer are initialized by Kaiming initialization [192]. The initial learning rate is set to 0.001 for NLU, and 0.0001 DST, NLG, respectively. We use the Adafactor [163] optimizer and the training batch size is set to 64 on Nvidia A10 GPUs. We tried a wide range of learning rates from 1e-2 to 1e-6 then set the initial training rate to 1e-4 in all training. Our code is developed based on *Soft-Mixture-of-Experts*³ and TOA-

³<https://github.com/fkodom/soft-mixture-of-experts.git>

TOD⁴. Code repository will be released to the public soon.

Because different batch sizes will result in different padded lengths, inference results are slightly changed by batch sizes due to Softmax over input tokens in the Soft-MoE layer. We make inferences on several selected batch sizes and report average scores. We found out that different batch sizes in our experiments have negligible influence on the inference results⁵.

Table 5.8: Joint Goal Accuracy (%) for DST on MultiWOZ 2.1 and 2.2. Results with ‡ are from He *et al.*(2022). † represents the results of our re-implementation. All others are reported as in the original papers.

Model	Pre-Trained Model	MultiWOZ2.1	MultiWOZ2.2
TRADE	-	45.6	45.4
TripPy	BERT-base	55.29	-
TripPy+SaCLog	BERT-base	60.61	-
CONVBERT-DG	BERT-base	55.29	-
SimpleTOD	DistilGPT-2	55.76	-
SOLOIST	GPT-2	56.85	-
AG-DST	PLATO-2	57.26	57.26
UniLM [‡]	UniLM	54.25	54.25
SPACE-3	UniLM	57.50	57.50
PPTOD _{base}	T5-base	57.10	-
PPTOD _{large}	T5-large	57.45	-
D3ST _{base}	T5-base	54.2	56.1
D3ST _{large}	T5-large	54.5	54.2
D3ST _{XXL}	T5-XXL	57.80	58.7
T5DST _{+desc}	T5-base	56.66	57.6
TOATOD _{small} [†]	T5-small	59.49	59.33
TOATOD _{base} [†]	T5-base	59.51	60.02
SMETOD _{small}	T5-small	59.69	59.60
SMETOD _{base}	T5-base	60.36	60.08

⁴<https://github.com/sogang-isds/TOATOD.git>

⁵We conducted a hypothesis test and found out p-value < 0.01 for scores changed by batch size. Statistics are summarized in section C.1.

Table 5.9: Evaluation of NLG on Inform, Success, BLEU, and Combined Scores, where Combined = (Inform + Success) \times 0.5 + BLEU.

Model	Backbone	MultiWOZ2.1				MultiWOZ2.2			
		Inform	Success	BLEU	Combined	Inform	Success	BLEU	Combined
DOTS	BERT-base	86.65	74.18	15.90	96.32	-	-	-	-
DiactTOD	S-BERT	-	-	-	-	89.5	84.2	17.5	104.4
SimpleTOD	DistilGPT-2	85.00	70.50	15.23	92.98	-	-	-	-
SOLOIST	GPT-2	-	-	-	-	82.3	72.4	13.6	90.9
UBAR ^Δ	GPT-2	95.70	81.80	16.50	105.25	83.4	70.3	17.6	94.4
MinTL ^Δ	BART _{large}	-	-	-	-	73.7	65.4	19.4	89.0
RewardNet ^Δ	BART _{large}	-	-	-	-	87.6	81.5	17.6	102.2
GALAXY	UniLM	95.30	86.20	20.01	110.76	85.4	75.7	19.64	100.2
PPTOD _{base}	T5-base	87.09	79.08	19.17	102.26	-	-	-	-
Mars	T5-small	-	-	-	-	88.9	78.0	19.9	103.4
MTTOD [‡]	T5-base	90.99	82.08	19.68	106.22	85.9	76.5	19.0	100.2
RSTOD [‡]	T5-small	93.50	84.70	19.24	108.34	83.5	75.0	18.0	97.3
TOATOD _{small}	T5-small	92.10	80.40	18.29	104.54	85.80	74.00	18.00	97.90
TOATOD _{base}	T5-base	97.00	87.40	17.12	109.32	90.00	79.80	17.04	101.94
KRLS	T5-base	-	-	-	-	89.2	80.3	19.0	103.8
SPACE-3	-	95.30	88.0	19.30	110.95	-	-	-	-
SMETOD _{small}	T5-small	91.80	79.30	16.58	102.13	89.6	76.2	17.1	100.1
SMETOD _{base}	T5-base	92.30	78.80	16.88	102.43	89.0	76.0	17.6	99.7

5.3.4 Results & Discussion

We show the effectiveness of our models on **NLU** (subsection 5.3.4), **DST** (subsubsection 5.3.4), and **NLG** (subsubsection 5.3.4) in task-oriented dialogue systems compared to plenty of strong baselines. In the experiments, we fine-tune SMETOD using the small and base versions of PPTOD[142], which continues pre-training T5 [105] on large dialogue corpora, as the start point. We observe that SMETOD is state-of-the-art on NLU and DST and comparable with existing baselines on NLG. We also study the improvement of efficiency with SMETOD (subsection 5.3.5). In subsection 5.3.6, we investigate model performance when the Soft-MoE layers are in different architectures.

i. Intent Prediction

The goal of intent prediction, known as NLU in a task-oriented dialogue system, is to identify the user’s intention based on the user’s utterance. We conduct experiments on

three benchmarks: Banking77 [185], CLINC150 [186], and HWU64 [187]. We report Accuracy (%) of predicting an intention correctly for evaluation.

Baselines. Baselines have a wide range from BERT-based models: CONVBERT [193], CONVERT [185], UniLM-based models: SPACE-2 [194], SPACE-3 [102], to T5-based TOATOD [99]. All baseline models utilizing BERT and UniLM follow a classification-based approach, employing a classifier featuring a Softmax layer to make predictions from a predefined set of intents.

Evaluation Results. Table 5.7 shows that our approaches perform state-of-art on CLINC150, which has the most number of intent types, where ^b means the NLG results on MultiWOZ 2.1 is from [195]. ^Δ shows models that require oracle dialogue states for prediction. [#] are obtained from DialogGLUE leaderboard⁶. On the other two benchmarks, our approaches have the highest accuracy compared to other generation-based approaches. Classification-based approaches are better which may benefit from smaller numbers of intents to choose from. Compared to classification models, SMETOD copes with the classification task as a generation problem by directly generating the text label. Therefore, when adapting to a new classification task, SMETOD is more scalable to new domains and tasks and can predict intents that are not in the ontology.

ii. Dialogue State Tracking

As a crucial component in task-oriented dialogue systems, DST determines the user goals based on the history of dialogue turns. For the evaluation of DST models, we use joint goal accuracy (JGA) which is the average accuracy of predicting all slot-values for the current turn correctly.

Baselines. In Table 5.8, we compare SMETOD with a wide range of classification-based approaches: TRADE [58], TripPy [90], TripPy + SaCLog [196], CONVBERT-DG [193], SimpleTOD [13], SOLOIST [101], AG-DST [197], SPACE-3 [102], and generation-based

⁶<https://eval.ai/web/challenges/challenge-page/708/leaderboard/1943>

approaches: PPTOD [142], D3ST [198], T5DST [112], and TOATOD [99]. Except for results with special symbols, All other results are from MultiWOZ leaderboards⁷.

Evaluation Results. Compared to other approaches, SMETOD obtains state-of-the-art JGA on MultiWOZ 2.1 and 2.2 among all generation-based approaches. Our model is more flexible to generate slot-value pairs while classification-based models are limited to the pre-defined ontology. The results show that our model can benefit from not only the transfer learning capacities of per-trained models but also the improvement of model size.

iii. End-to-End Response Generation

End-to-end dialogue response generation, aiming at evaluating the model in the most realistic, fully end-to-end setting, where the generated dialogue states are used for the database search and response generation [13, 142], is NLG in task-oriented dialogue system. Our models evaluated on MultiWOZ generates responses not only related to the dialogue history but also grounded on the database (DB) state.

Metrics. For evaluation, we follow the individual and combined metrics in [13]: Inform, Success, and BLEU, and Combined score which is defined as $\text{Combined} = (\text{Inform} + \text{Success}) \times 0.5 + \text{BLEU}$. Specifically, Inform rate measures the correctness of entities in the response. Success rate success rate assesses attribute fulfillment requested by user. BLUE score is used to measure the fluency of the generated responses.

Baselines. In Table 5.9, we compare our model with several strong baselines: DOTS [199], DiactTOD [200], SimpleTOD [13], SOLOIST [101], UBAR [201], MinTL [202], Reward-Net [203], GALAXY [204], PPTOD [142], RSTOD [195], MTTOD [205], TOATOD [99], KRLS [206].

Evaluation Results. On both MultiWOZ 2.1 and 2.2 datasets, SMETOD performs, though not the best, comparable to T5-based models except TOATOD_{base}. We hypothesize that metrics hinder each other from being improved together and may require a mechanism to

⁷<https://github.com/budzianowski/multiwoz>

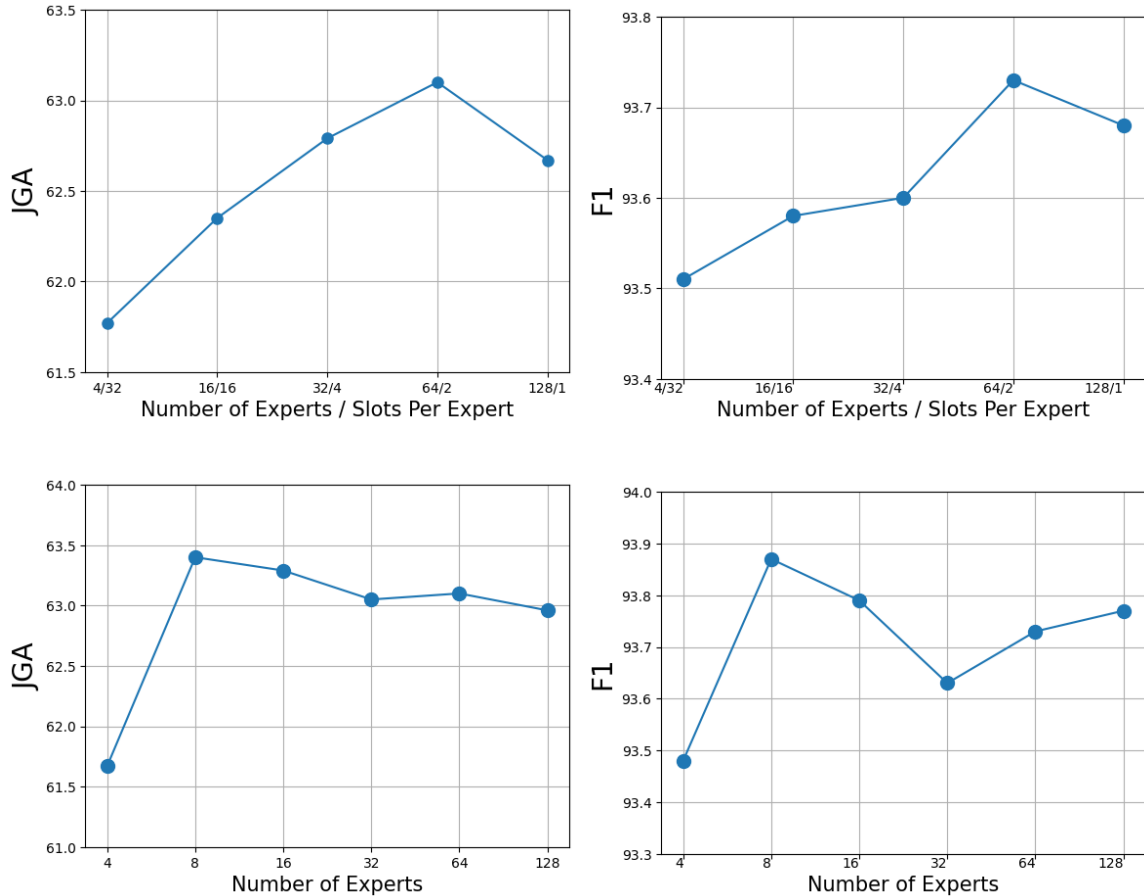


Figure 5.4: **(Top)** Performance of SMETOD as a function of the number of experts, for models with a fixed number of experts \times slots-per-expert. **(Bottom)** Performance of SME-TOD trained with increased experts and 2 slots per expert. JGA and F1 scores are on MultiWOZ 2.1 dev set for DST.

promote performance towards specific metrics, for example, REINFORCE [207]. Besides, we observe that only replacing the Feed-Forward layer in Transformer encoders as in [182] without copying weights to experts doesn't generate the best results in our dialogue system. It might be because their implementation requires a large amount of data to pre-train, which is inappropriate in the task-oriented scenario. It demonstrates that by duplicating pre-trained weights and fine-tuning, SMETOD optimizes well for DST and NLG, respectively, maintaining the prior knowledge learned from the pre-trained model. To ensure statistical significance, we report the results in Table 5.8 and Table 5.9 as the average over five random seeds. Additional details can be found in section C.1.

Table 5.10: Comparison of the inference time with small and base-size models of PPTOD and TOATOD for NLG on MultiWOZ 2.1. All models are experimented with 5 same and randomly sampled batch sizes. Average time is reported. ↓: Smaller is better.

Model	Small↓	Base↓
PPTOD	1×	3.163×
TOATOD	1.116×	3.519×
SMETOD	1.005×	3.095×

5.3.5 Time Complexity Analysis

According to [182], the time complexity of the Soft-MoE layer can be reduced to $O(l^2d + lk)$, given input token length l , model hidden dimension d , and the cost of applying an expert per token $O(k)$. Thus, the time complexity is constant and the same as the single-headed self-attention cost by increasing the number of experts m and scaling slots per expert $p = O(l/m)$ accordingly, which will not become a bottleneck in Transformer.

We show in Table 5.10 that SMETOD could make inferences without bringing about much extra time. SMETOD_{small} is 3.5 times larger than PPTOD and TOATOD while achieving a similar inference speed as the former. Our SMETOD_{base} has even less inference time while its model size is 4 times of PPTOD_{base} . It proves that we can achieve much better scaling while cost is roughly constant [182], with the benefit of improved performance.

5.3.6 Impact of Expert Numbers

We investigate the impact of expert and slot numbers in our models on the development set of MultiWOZ 2.1 for DST as illustrated in Figure 5.4. First, we fix the total number of slots to 128 and vary expert numbers $\{4, 16, 32, 64, 128\}$ by scaling slot numbers per expert. Results suggest the best configuration is 64 experts and 2 slots per expert. Then, we set the number of slots per expert to one and evaluate performance with regard to the number of experts. The number of experts 8 and 16 perform better than others. It should be mentioned that the model size scales with increasing expert numbers only. Meanwhile, we

observe performance is not always increasing with the number of experts, indicating there is a trade-off between model size and the amount of training data.

SMETOD scales with increased experts. The best configurations are 64 experts, at two slots per expert. The next question is how many experts to use. When increasing the number of slots, the performance grows first and then decreases modestly, with optimal at the 8 experts. We also conduct an ablation on the placement of SoftMoE layer in T5, as in section C.2.

5.4 Conclusion

The incorporation of systematic knowledge in task-oriented dialogue systems represents a significant advancement in the field of IA. Through this chapter, we have demonstrated how structured approaches to knowledge integration can substantially improve the performance, scalability, and efficiency of dialogue systems.

Our Act-Aware Slot-Value Predicting (ADST) model combine dialogue acts, dialogue context and domain ontology, and leverages reading comprehension approaches to predict slots for both categorical and non-categorical slots, showing that attentions on both dialogue acts and dialogue context not only improve the joint goal accuracy, but also expand capacities of dialogue systems on reasoning how dialogue states are guided and transformed. We propose an efficient Soft Mixture-of-Experts (SMETOD) framework to satisfy requirements on diverse outputs in task-oriented dialogue systems. We demonstrate that incorporating Soft-MoE to our dialogue system achieves remarkable success on Multi-WOZ baselines and optimizes outputs of each submodule, showing it powerful technique for task-oriented dialogue systems with better scaling performance while maintaining time efficiency.

The success of systematic knowledge incorporation in improving dialogue system performance while maintaining scalability demonstrates its fundamental importance in advancing the field. As task-oriented dialogue systems continue to evolve and find new appli-

cations, the principles and approaches outlined in this chapter will serve as valuable guidelines for future research and development efforts in creating more sophisticated, reliable, and user-friendly conversational interfaces.

CHAPTER 6

ENHANCING COMMONSENSE REASONING ACROSS LANGUAGES

6.1 Introduction

TOD systems are increasingly applied across various industries, including customer service, healthcare, and education. Despite their growing adoption, a notable gap remains between current system performance and their application scenarios. This gap is particularly evident in scenarios requiring nuanced understanding of context, cultural references, or language barrier. To address these challenges, we introduce cross-lingual knowledge transfer as a means of enhancing CSR for low-resource languages, thereby improving downstream dialogue applications.

Commonsense reasoning (CSR) relies on shared and unchanged knowledge across different languages and cultures to help computers understand and interact with humans naturally [208]. CSR is a crucial problem in natural language processing that has been proved important for AI systems [209, 210]. The study of CSR has a long tradition in English as a way to probe the capacities of language models on reasoning about natural language [211, 212, 213, 209, 214]. Recently, several multilingual CSR datasets are proposed to support a series of multilingual CSR tasks, such as commonsense coreference resolution [215, 216], causality reasoning [217], commonsense question-answering [218] and sentence completion [218]. Multilingual pre-trained language models [219, 220, 125, 128] proved effective in various cross-lingual tasks with their contextual representations. The most related work to ours is XCSR [218] which proposes multilingual contrastive pre-training (MCP) and evaluates on XCSQA and XCODAH.

Cross-lingual CSR aims to reason commonsense across languages, which is the key to bridging the language barrier in natural language understanding and generalizing CSR

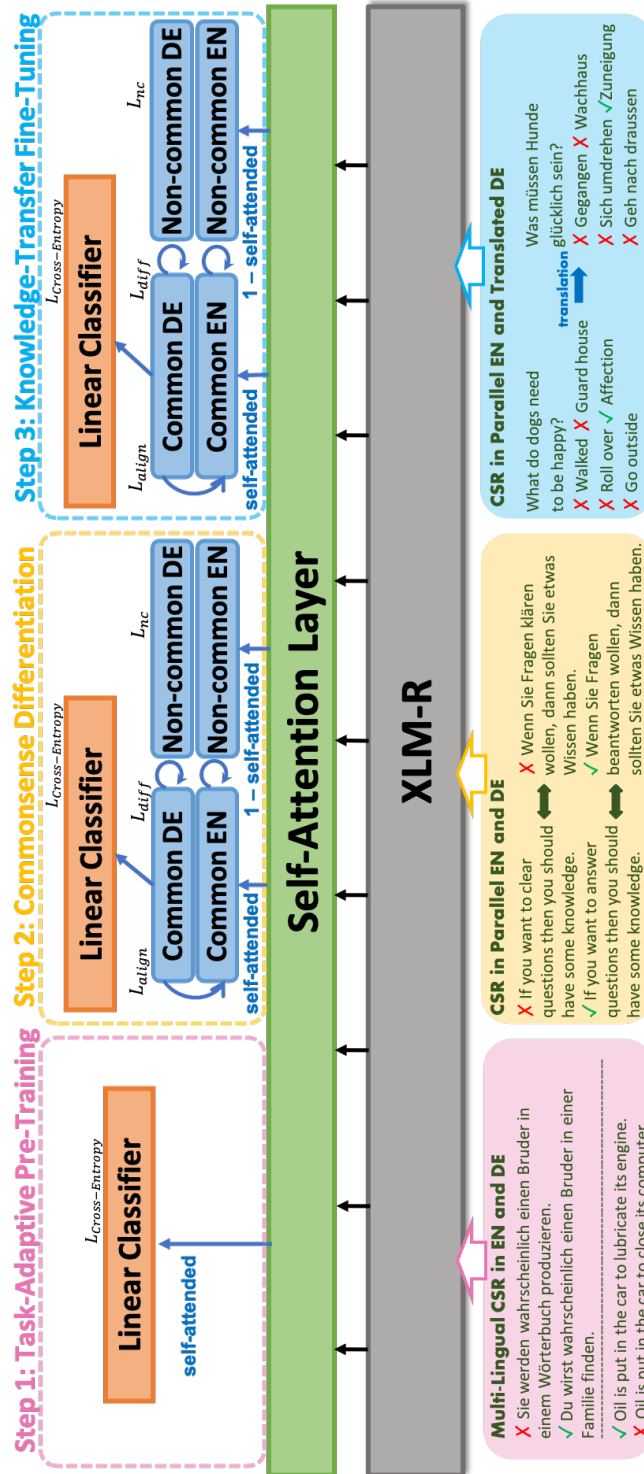


Figure 6.1: The overview of CLICKER framework, which is trained in three steps with joint objectives for cross-lingual commonsense reasoning. Parameters in XLM-R and self-attention layers are shared in all steps. Each ✓ or ✗ represents whether the following choice is commonsense reasonable or not, respectively.

to a broader scope [221]. Recently, several cross-lingual datasets are proposed amidst the surging interest in cross-lingual CSR, e.g. XCOPA [217], X-CSQA [218], and X-CODAH [218]. Multilingual pre-trained language models (mPTMs) based on the Transformer [119], such as mBERT [219], XLM [220], XLM-R [125] and InfoXLM [222], have also been demonstrated to have potentials for CSR in multiple languages [223, 221, 217, 218]. The performance of mPTMs for non-English CSR, however, is typically worse than that for English CSR due to the lack of non-English data for training [224, 125, 217]. Furthermore, mPTMs have raised concerns about their ability to transfer commonsense knowledge across languages, as they are not capable of 1) differentiating between commonsense and non-commonsense knowledge and 2) improving the CSR for any specific language in multilingual scenarios [125].

To address the above issues, we propose **Cross-LIngual Commonsense Knowledge transfER** (CLICKER) to bridge the performance gap of using mPTMs for CSR between the source (**English**) and the target (**non-English**) language by eliciting commonsense knowledge explicitly via cross-lingual task-adaptive pre-training [225]. Specifically, CLICKER is a three-step framework based on XLM-R [125]. First, we conduct task-adaptive pre-training on the multilingual commonsense corpora to enable XLM-R to perform the CSR task better. In this process, the self-attention [119] mechanism is adopted to obtain multilingual embeddings for CSR. Second, we distinguish between commonsense and non-commonsense knowledge by jointly optimizing their similarities with bilingual and parallel data. Third, the extracted commonsense knowledge representation is further fine-tuned on the downstream cross-lingual CSR tasks.

Experimental results demonstrate that our approach significantly reduces the performance discrepancies between English and German on CSR. Moreover, it outperforms XLM-R baselines on both X-CSQA and X-CODAH benchmarks [218]. Further analysis indicates that CLICKER can extract cross-lingual commonsense representations more effectively, and with better interpretability.

6.2 Attention-Based Cross-Lingual Commonsense Knowledge Transfer

Knowledge graph construction serves as a fundamental component in modern AI models, particularly in ToD systems. Drawing from CLICKER’s approach to cross-lingual knowledge representation, ToD systems can benefit from structured knowledge representations that capture both task-specific information and general conversational knowledge. These knowledge graphs can represent entities, relationships, and contextual information crucial for meaningful dialogue interactions.

The embedding of knowledge into neural networks follows principles similar to CLICKER’s attention-based architecture[39]. In ToD systems, this involves creating dense representations of knowledge that can be efficiently accessed and utilized during dialogue processing. The attention mechanism, as demonstrated in CLICKER, proves particularly valuable in selecting relevant knowledge based on the current dialogue context. Transfer learning and multi-task learning approaches become especially relevant when dealing with multiple domains or languages in ToD systems. CLICKER’s success in transferring commonsense knowledge across languages provides a blueprint for transferring dialogue-related knowledge across different tasks and domains. This transfer can significantly improve performance in low-resource scenarios or new domains where training data is limited.

introduces the CLICKER model based on XLM-R [125] for cross-lingual CSR. As illustrated in Figure 6.1, CLICKER extracts commonsense knowledge from English CSR to help non-English CSR¹ in three steps: 1) task-adaptive pre-training, 2) commonsense differentiation, and 3) knowledge-transfer fine-tuning.

6.2.1 Problem Definition

The CSR task aims to select one from multiple choices that are most reasonable in commonsense given the previous statement or question. For example, the plausible choice of

¹In this paper, we take German as an example of a foreign language that is not up to par with English for CSR.

answer to “*What is a great place to lay in the sun?*” is “*beach*” rather than “*in the basement*” or “*solar system*”. Denoting a set of choices for CSR as $\mathbf{S}^{(j)}$'s, $j \in [1, \dots, |C|]$, where the number of choices for each input as $|C|$, the goal is to predict the common sense choice:

$$\tilde{y} = \operatorname{argmax}_j P(y = j | \{\mathbf{S}^{(1)}, \dots, \mathbf{S}^{(|C|)}\}) \quad (6.1)$$

6.2.2 Step One: Task-Adaptive Pre-Training

Task-adaptive pre-training uses self-attention to enable the Transformer in XLM-R to learn representations for CSR. Specifically, the input is a tokenized utterance, i.e. $\mathbf{S}_i = \{[\text{CLS}], q_i^{(1)}, \dots, q_i^{(k)}, [\text{SEP}]\}$, where $i \in [1, \dots, N]$, N is the size of the dataset, and k is the length of sequence. A self-attention layer is built on top of the Transformer to obtain attentions toward commonsense knowledge from the Transformer’s pooled output states. The self-attended outputs are optimized by the Cross-Entropy loss through a multiple-choice classifier to select commonsense-reasonable choices. Our model is trained on multilingual CSR datasets containing examples of both English (**EN**) and German (**DE**).

6.2.3 Step Two: Commonsense Differentiation

In this step, the representation of commonsense knowledge shared across languages is differentiated from the non-commonsense representation using EN and DE datasets in parallel. The inputs are similar to those in subsection 6.2.2, while inputs with the same semantics in different languages are mapped together. We note here that the parallel datasets are not necessarily restricted to CSR datasets, but can be generalized to any bilingual datasets for mapping semantics of English and the non-English language, e.g. bilingual dictionaries or textbooks.

The output states of the Transformer are pooled and weighted by the self-attention layer followed by a linear projection, being extracted as **commonsense** embeddings X_i and **non-commonsense** embeddings \tilde{X}_i , respectively.

$$X_i = FFN(\text{Attention}(\mathbf{O}_i)) \quad (6.2)$$

$$\tilde{X}_i = FFN(1 - \text{Attention}(\mathbf{O}_i)) \quad (6.3)$$

where \mathbf{O}_i are output hidden states from the last layer of the Transformer for the i -th input, and FFN represents a *Feed-Forward* layer. For brevity, we omit index i in the following equations.

We use X^{EN} and X^{DE} to denote commonsense embeddings of English and German inputs, respectively. Similarly, \tilde{X}^{EN} and \tilde{X}^{DE} represent non-commonsense embeddings. Knowledge mapping is made by measuring the similarities between commonsense embeddings and non-commonsense embeddings. Specifically, we maximize cosine similarity between English and German embeddings that share the same and valid commonsense knowledge, i.e. $X^{EN_{j^*}}$ and $X^{DE_{j^*}}$, as in Eq. (Equation 6.4). And we minimize cosine similarity between $X^{EN_{j^*}}$ and X^{EN_j} , as in Eq. (Equation 6.5). j^* is the index of the choice that is reasonable in commonsense, $j \in [1, \dots, |C|]$ and $j \neq j^*$. Such that similar commonsense knowledge in both languages is projected into the same position in semantic representation space.

$$\mathcal{L}_{align} = 1 - \cos(X^{EN_{j^*}}, X^{DE_{j^*}}) \quad (6.4)$$

$$\begin{aligned} \mathcal{L}_{diff} = & \sum_{j=1, j \neq j^*}^{|C|} (\max(0, \cos(X^{EN_{j^*}}, X^{EN_j})) \\ & + \max(0, \cos(X^{DE_{j^*}}, X^{DE_j}))) \end{aligned} \quad (6.5)$$

On the other hand, non-commonsense embeddings represent knowledge unrelated to cross-lingual commonsense. Assuming that the correct choice and other incorrect choices associated with the same question share similar non-common-sense knowledge, we maximize the intra-language cosine similarity of non-commonsense embeddings. Moreover, the correct choice of different languages should share the same non-commonsense knowledge so that we maximize inter-language cosine similarity jointly, as defined in Eq. (Equa-

tion 6.6).

$$\begin{aligned}
 \mathcal{L}_{nc} = & \sum_{j=1, j \neq j^*}^{|C|} (1 - \cos(\tilde{X}_i^{EN_{j^*}}, \tilde{X}_i^{EN_j})) \\
 & + \sum_{j=1, j \neq j^*}^{|C|} (1 - \cos(\tilde{X}_i^{DE_{j^*}}, \tilde{X}_i^{DE_j})) \\
 & + 1 - \cos(\tilde{X}_i^{EN_{j^*}}, \tilde{X}_i^{DE_{j^*}})
 \end{aligned} \tag{6.6}$$

All the losses above and the Cross-Entropy loss are optimized as the joint training objective for cross-lingual CSR. We use output commonsense embeddings $X^{DE_{j^*}}$ and X^{DE_j} to calculate the Cross-Entropy loss.

6.2.4 Step Three: Knowledge-Transfer Fine-Tuning

Finally, our model is fine-tuned by the training objectives similar to subsection 6.2.3 for evaluating CSR on the multiple-choice question-answering (QA) and the clause-selection tasks, leveraging parallel CSR datasets of English (EN) and German translated from English (EN_DE) as inputs. Different from previous steps, each input of XLM-R is the concatenation of a question and a choice of answer which are then split into tokens with additional special ones, i.e. $S_i = \{[\text{CLS}], q_i^{(1)}, \dots, q_i^{(m)}, [\text{SEP}], [\text{CLS_Q}], a_i^{(1)}, \dots, a_i^{(n)}, [\text{SEP}]\}$, where [CLS_Q] is the beginning special token of the answer spans, q_i and a_i are tokens of the question and answer, and m, n are numbers of question and answer tokens, respectively.

6.3 Experiments and Analyses

We use English and German subsets of Mickey Corpus [218] for Step 1 (subsection 6.2.2) to warm up the pre-trained XLM-R_B [125] for cross-lingual CSR tasks. Mickey Corpus is transformed into multiple-choice problems with 5 natural-language choices, and the goal is to select the most commonsense plausible one. This corpus is split into 16k examples for training and 1k examples for evaluation for each language. Then we leverage parallel datasets in Mickey Corpus again in Step 2 (subsection 6.2.3) to obtain semantic

mappings of English and German and differentiate commonsense and non-commonsense embeddings. In Step 3 (subsection 6.2.4), the CLICKER model is fine-tuned and evaluated on X-CSQA and X-CODAH, which are evaluation benchmarks for mPTMs on cross-lingual CSR, both containing $\sim 8k$ English training examples and ~ 500 validation and test examples in other 15 languages. In order to obtain the parallel training sets in English and German to use in our models, we leverage machine translation to process the English training set in X-CSQA and X-CODAH to German with MBART [120]. To make fair comparisons, our experiments use English and machine-translated German for training, and use German for evaluation on both benchmarks.

Our models are based on XLM- R_B [125]. Implementations are based on Hugging-Face [226] and follow initializations in [218]. Each input is split into tokens by Word-Piece [227] with special tokens added. The optimizer is AdamW [169] with a linearly-decaying scheduler, and the warm-up steps are 100. We run experiments on NVIDIA V100 up to 8 GPUs.

We compare our model with the multilingual contrastive pre-training (MCP) [218] model based on XLM- R_B . MCP model is trained on permuted Mickey Corpus for multilingual contrastive training and fine-tunes on cross-lingual CSR training set in English only. Instead, we re-implement it to train on the combination of English and German Mickey Corpus. Then we fine-tune it on both English and machine-translated German CSR training sets and evaluate it on the test set in German to make a fair comparison with our method.

The following subsections describe the experimental results and analyze CLICKER models on the cross-lingual CSR benchmarks X-CSQA and X-CODAH in German. Note that our experiments are conducted for commonsense knowledge transfer from English to German, but the approach can be extended to other languages.

Table 6.1: Accuracy on the test set of X-CSQA and X-CODAH in German. MCP(XLM- R_B)* model is trained in English and machine-translated German. The *align*, *diff*, *nc* refer to the objectives in equation (Equation 6.4), (Equation 6.5), and (Equation 6.6), respectively.

Models	Acc
<i>X-CSQA</i>	
MCP(XLM- R_B)* [218]	48.8
CLICKER - <i>base</i>	49.6 (+0.8)
CLICKER - <i>align</i>	50.6 (+1.8)
CLICKER - <i>align+diff</i>	50.6 (+1.8)
CLICKER - <i>nc</i>	49.8 (+1.0)
CLICKER - <i>align+nc</i>	49.6 (+0.8)
<i>X-CODAH</i>	
MCP(XLM- R_B)* [218]	49.2
CLICKER - <i>base</i>	50.2 (+1.0)
CLICKER - <i>align</i>	49.6 (+0.4)
CLICKER - <i>align+diff</i>	50.3 (+1.1)
CLICKER - <i>nc</i>	50.8 (+1.6)
CLICKER - <i>align+nc</i>	49.6 (+0.4)

6.3.1 Experimental Results

Table 6.1 shows the test accuracy of baselines and CLICKER models for CSR in German. Different combinations of losses are applied in experiments for optimizing commonsense differentiation. We observe consistent improvements with our three-step framework by extracting commonsense knowledge with self-attentions (i.e. CLICKER - *base*) on both datasets compared to baselines.

Results show that the *align* loss further improves the base CLICK model on X-CSQA. And the *non-commonsense* (*nc*) loss is proved effective on both datasets. The best performance on X-CSQA is achieved when using the *align* loss with or without the *diff* loss, which shows that lining up embeddings in English and German with the same commonsense knowledge dominates the performance of CSR. Besides, the model with *align* and *nc* loss is slightly inferior to the model with *nc* loss only on X-CSQA. On X-CODAH, our CLICK models perform the best with the *nc* loss which maximizes the cosine similarity of non-commonsense embeddings, improving 1.6% on accuracy.

6.3.2 Discussion

Our models address the alignment of extracted embeddings with different combinations of objectives. The fact that *align+nc* loss is not as good as *nc* loss alone suggests a conflict between aligning the commonsense embeddings and aligning the non-commonsense embeddings. This can be explained as both objectives aiming to maximize the cosine similarity of embeddings, making it harder for the model to discern different commonsense knowledge in them. From the best accuracy achieved on two datasets, we conjecture the quality of commonsense embeddings (optimized by *align* and *diff* losses) dominates CSR on X-CSQA, while non-commonsense embeddings (optimized by *nc* loss) dominates that on X-CODAH. The reason for this may be extracting commonsense knowledge for clause selection in X-CODAH is more challenging than multiple-choice QA in X-CSQA, whereas separating the non-commonsense embeddings help the multiple-choice classifier understand the commonsense portion with less noise. We also observe that using *align* and *nc* losses together is not the best practice in our experiments on both datasets. It suggests that jointly optimizing both objectives makes it more difficult for the multiple-choice classifier to predict correctly, as correct choices are pushed closer to incorrect ones.

Commonsense v.s. Non-commonsense. To investigate the effectiveness of our learned commonsense embeddings, we evaluate the accuracy predicted by commonsense embeddings or non-commonsense embeddings on the dev set of X-CSQA. As seen in Table 6.2, the performance of commonsense embeddings is significantly better than that of non-commonsense embeddings. It is as expected, as our models are fine-tuned by cross-lingual CSR objectives to discern commonsense embeddings, while maximizing the similarity of non-commonsense embeddings. Non-commonsense embeddings can induce confusion for CSR, such that combining both embeddings performs worse than using commonsense embeddings only.

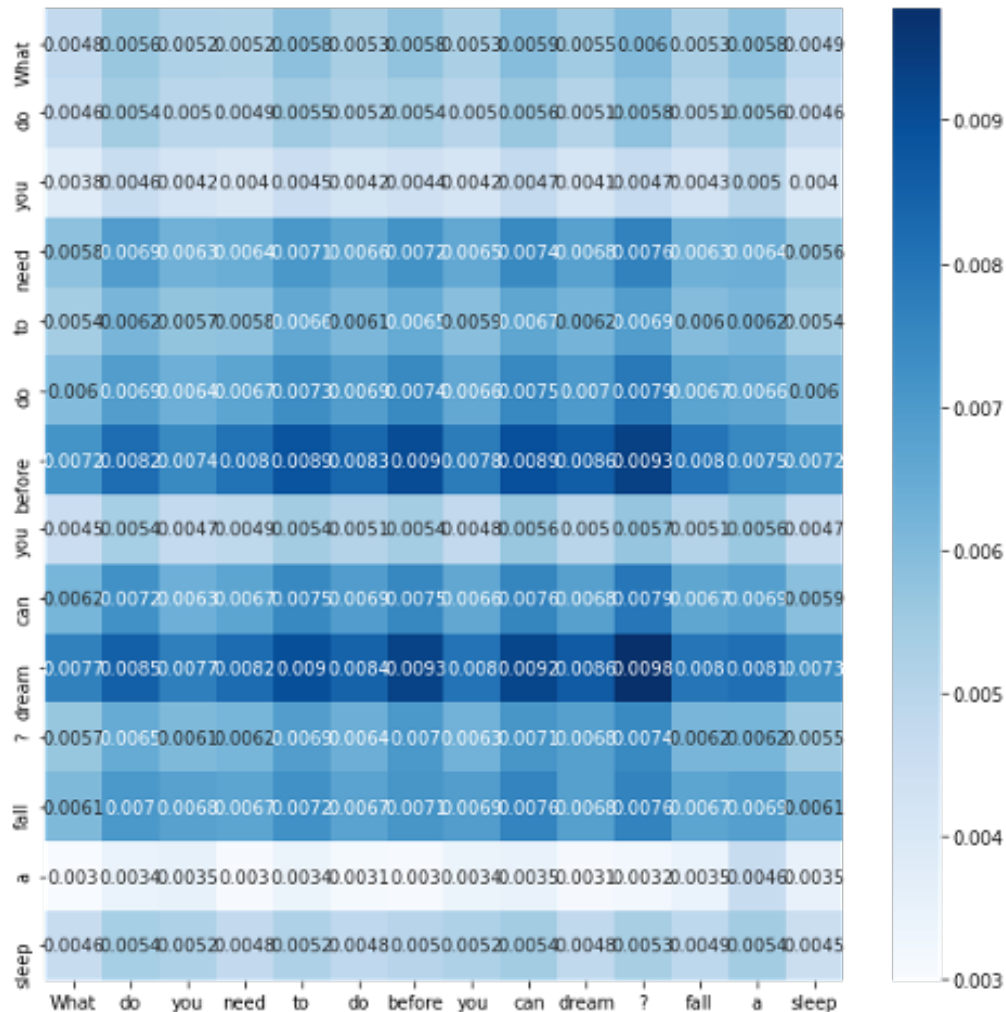


Figure 6.2: Attention head of the self-attention layer. The given example is from X-CSQA.

6.3.3 Attention Heatmap from X-CODAH

Does self-attention imply commonsense knowledge? We suppose that self-attentions in our models can effectively attend to tokens that are essential for commonsense plausibility. Figure 6.2 is the heatmap of the attention head in the self-attention layer evaluated on an example “*What do you need to be before you can dream?*” from X-CSQA. It’s noteworthy to see that attention weights are given much more to commonsense-related tokens, such as “*before*”, “*dream*” and “*sleep*” tokens. A similar phenomenon is observed on X-CODAH as well. These tokens are weighted to generate commonsense embeddings and help our model improve accuracy and interpretability of reasoning commonsense knowledge.

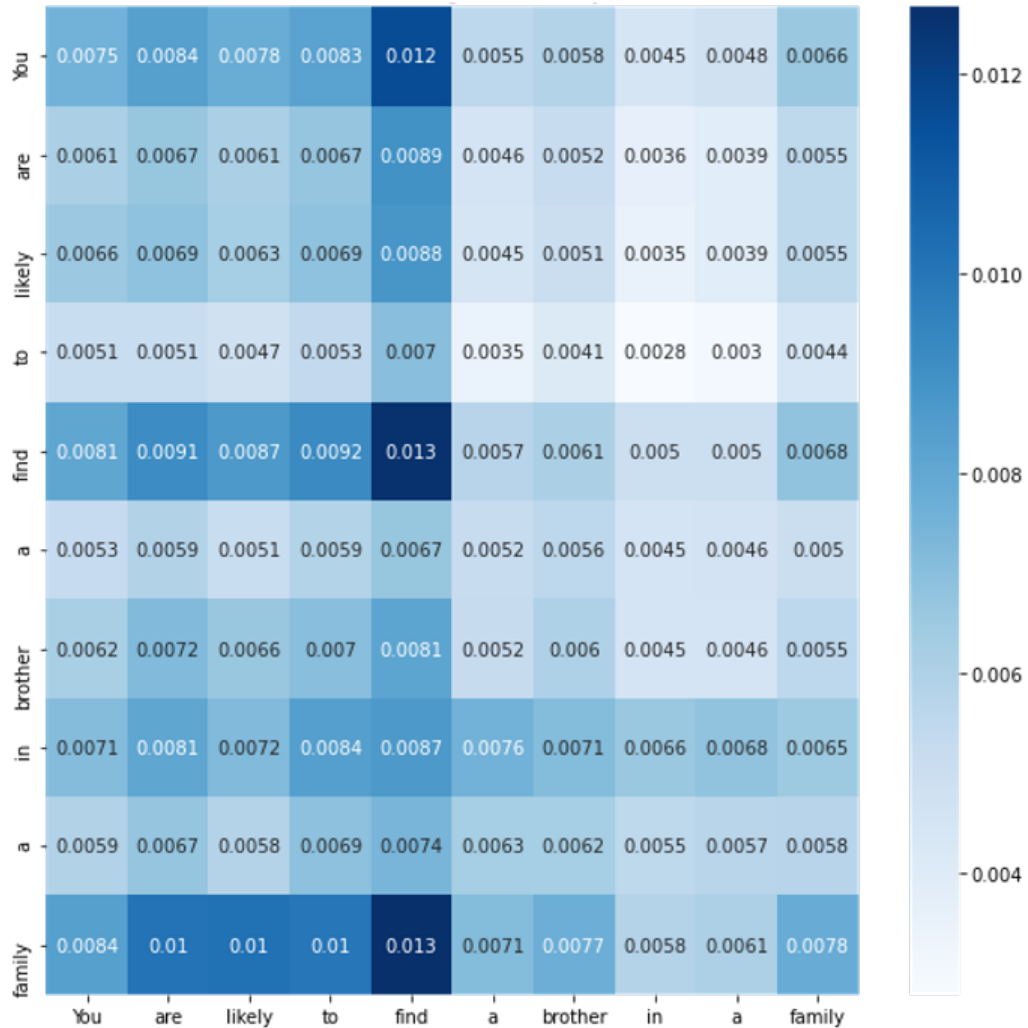


Figure 6.3: Attention head of the self-attention layer. The given example is from X-CODAH.

Classifier Input	Dev Acc
Commonsense	47.8
Non-Commonsense	11.0
Commonsense + Non-Commonsense	47.6

Table 6.2: Dev accuracy on X-CSQA taking commonsense or non-commonsense embeddings as inputs for the classifier.

6.4 Conclusion

In this paper, we propose a cross-lingual framework CLICKER for commonsense reasoning. Experiments on X-CSQA and X-CODAH demonstrate the effectiveness of CLICKER

in cross-lingual commonsense reasoning as it not only reduces performance discrepancies of commonsense reasoning between English and non-English languages but also improves the interpretability of commonsense knowledge across languages. The possibility of extending our approach to other low-resource languages will be beneficial for alleviating data scarcity in cross-lingual commonsense reasoning.

Knowledge integration in ToD systems represents a crucial advancement in human-machine interaction. The success of approaches like CLICKER in cross-lingual knowledge transfer demonstrates the potential for similar breakthroughs in task-oriented dialogue systems. As these systems continue to evolve, the integration of external knowledge sources will become increasingly sophisticated, enabling more natural, effective, and contextually aware conversations.

The future of human-machine conversational agents lies in their ability to seamlessly combine task-specific knowledge with broader contextual understanding. By building on current successes in knowledge-augmented learning and addressing ongoing challenges, ToD systems will continue to narrow the gap between human-human and human-machine interactions, ultimately leading to more effective and satisfying user experiences.

6.5 Conclusion

In this chapter, we propose a cross-lingual framework CLICKER for commonsense reasoning. The main idea of this approach is to separate commonsense and non-commonsense embeddings and extract commonsense knowledge across languages. Specifically, we adopt self-attentions to draw embeddings for commonsense reasoning, and then jointly optimize the similarity distances to align or distinguish commonsense and non-commonsense knowledge. Experiments on X-CSQA and X-CODAH demonstrate the effectiveness of CLICKER in cross-lingual commonsense reasoning as it not only reduces performance discrepancies of commonsense reasoning between English and non-English languages but also improves the interpretability of commonsense knowledge across languages. The pos-

sibility of extending our approach to other low-resource languages will be beneficial for alleviating data scarcity in cross-lingual commonsense reasoning.

CHAPTER 7

CONCLUSION

7.1 Summary of Contributions

This thesis investigates the role of knowledge in TOD systems for IA on enhancing the understanding of user goals and enabling task completion at scale, highlighting our contributions to the development of this field which has evolved rapidly during the deep learning era.

In chapter 2, we walked through the history of computational discourse analysis and dialogue systems, which dates back to the 1970s. At the time, researchers recognized the importance of these systems as a means of testing the language understanding capabilities of computer programs and managing service dialogues. However, it was not until the 2000s that dialogue systems began to play a significant role in the business world, driven by advancements in web technologies and customer service applications. Since 2017, the field has been completely transformed by the advent of large-scale supervised datasets and advancements in NLU, DST, and NLG models. Innovations in developing better datasets and more effective models contributes to the evolution of the field. We also explore the distinctions between open-domain and task-oriented dialogues, emphasizing the importance of addressing knowledge integration in TOD systems. Specifically, we described two primary types of knowledge—domain - specific and dialogue-level knowledge — highlighting their respective applications.

In chapter 3, we presented the methodology for incorporating knowledge into TOD systems. Specifically, we focus on applying domain-level knowledge within the context of DST tasks (e.g., understanding slot-value pairs, schema relationships) using methods from semantic task frameworks (e.g., QA, schema modeling), and incorporating dialogue-

level knowledge into TOD systems to handle the meta-level needs of diverse dialogue tasks (e.g., intent prediction and response generation). We propose two key applications that effectively leverage domain-level knowledge in DST. CoFunDST utilizes domain-level knowledge to enhance slot-value prediction in task-oriented dialogue systems by integrating candidate choices as contextual input. Similarly, SHEGO employs domain-level schema knowledge, embedding it into prompts to improve domain adaptation, resulting in more accurate and efficient DST. In addition to domain-level knowledge, two applications prominently incorporate dialogue-level knowledge. The Act-Aware approach directly integrates dialogue-level knowledge by leveraging dialogue acts as explicit input, which guides slot-value predictions and facilitates reasoning about dialogue states. On the other hand, the SMETOD provides a broader framework, where dialogue-level knowledge is embedded within task optimization but is not explicitly highlighted as a central feature. Moreover, we propose extracting and transferring commonsense knowledge across languages as an effective strategy to enhance CSR in low-resource languages, thereby advancing the performance of downstream dialogue applications.

In chapter 4, we demonstrated how combining transfer learning with DST models enables scalable TOD systems capable of handling new domains with minimal or zero-shot data. The Choice-Fusion DST approach introduced in this chapter advances state tracking by explicitly modeling candidate choices and leveraging domain ontologies, achieving superior zero-shot performance while maintaining interpretability. Its fusion mechanism integrates dialogue history, domain schemas, candidate choices, and previous states, creating a robust foundation for state tracking. Additionally, schema graph-guided prompts effectively utilize domain schema structures for efficient multi-domain adaptation. By capturing relationships between slots and values in a graph-based representation, this method reduces the need for extensive fine-tuning, enhancing scalability for real-world applications. These approaches aim to improve learning efficiency, generalization across domains, and effective knowledge transfer across tasks.

In chapter 5, we explored incorporating dialogue-level knowledge into TOD systems. The ADST approach demonstrated that explicitly modeling dialogue acts as systematic knowledge significantly enhances dialogue state tracking accuracy, enabling better understanding of user intentions in complex, multi-turn conversations. This improvement provides a solid foundation for more reliable task-oriented dialogue systems. Similarly, the SMETOD architecture shows how modular expertise can be integrated with systematic knowledge to develop scalable and efficient systems. By dynamically utilizing specialized knowledge while maintaining computational efficiency, this approach addresses scalability challenges in TOD systems. These advancements pave the way for leveraging generalizable and dialogue-level knowledge to create sophisticated, reliable, and user-friendly conversational interfaces.

In chapter 6, we address the challenge of leveraging cross-lingual knowledge for commonsense reasoning, where a system must understand the semantics of a given query and answer questions arising in conversations. To tackle this, we developed CLICKER, a cross-lingual language model designed to transfer commonsense knowledge for lower-resource languages to effectively participate in question-answering conversations. Our approach establishes state-of-the-art (SOTA) performance on the XCSQA and XCODAH benchmarks for knowledge understanding and commonsense reasoning. We believe that these advancements in question answering and sentence completion will contribute to the development of better IA in the future.

7.2 Limitations of Study

This dissertation, while making significant contributions to task-oriented dialogue systems, faces several important limitations. These limitations include computational constraints, data characteristics, and comparative evaluations.

One of the limitations relates to input processing and computational resources. The models face practical constraints in handling long dialogue contexts, as increasing con-

text length demands substantially more computational resources to maintain performance. In chapter 4, for example, the CoFunDST model necessitates pretraining on a large QA dataset, which requires substantial computational resources. Similar constraints affect the SHEGO framework, where dialogue history truncation can lead to information loss in longer conversations. The computational resources needed for training these models, particularly when scaling up model sizes, may present challenges for widespread deployment. This becomes especially relevant when considering the practical implementation of these systems in resource-constrained environments.

Data imbalance presents another significant challenge across multiple aspects of our work. In the categorical slot prediction task of CoFunDST, we observed substantial variations in the number of candidate choices across different slots (ranging from two choices for "hotel-type" to fifteen for "attraction-type"), affecting the model's ability to incorporate choices effectively. Furthermore, the training datasets exhibit imbalances between multiple-choice and extractive examples, potentially biasing the model's performance toward extractive answer generation. The SHEGO and Act-aware Dialogue State Tracker (ADST) framework similarly suffers from domain imbalance in training data, potentially favoring high-resource domains at the expense of others. In the ?? implementation in chapter 5, we faced challenges related to varying input token lengths, which can lead to inconsistent inference with different batch sizes. While these variations showed minimal impact in our experiments, they require further investigation, particularly for applications with diverse example lengths.

Cross-lingual capabilities remain constrained by current limitations. In chapter 6, the models show promising results for specific language pairs but may not generalize effectively to languages with fundamentally different linguistic structures. This limitation is particularly relevant when considering the global applicability of task-oriented dialogue systems.

Finally, we acknowledge limitations in our comparative evaluations. The exclusion of

certain baseline comparisons, such as with large language models like GPT-4[25] or Gemini[228], was necessitated by factors including the impact of prompt quality on results, differences in training scenarios (full-shot versus zero/few-shot), and potential data contamination issues in public benchmarks. Additionally, the lack of comprehensive per-domain performance analysis on certain datasets limits our understanding of domain-specific effectiveness. Additionally, we acknowledge limitations in our evaluation scope, as our studies primarily focused on specific domains and datasets, potentially not fully representing the diverse range of real-world dialogue scenarios.

These limitations, while significant, also point to valuable directions for future research and improvement in task-oriented dialogue systems. They highlight the need for more robust and scalable solutions that can better handle the complexities and diverse requirements of real-world applications.

7.3 Discussions and Future Work

7.3.1 Transfer learning for large-scale task-oriented DST

The scarcity of DST data limits the performance of DST models for cross-domains and new domains. On the other hand, non-categorical DST has much more free-form values to be predicted and tends to generate more errors. With key values generated across domains, DST models are possible to get extra power from domains other than the previous dialogue context itself. We adopt this idea and compare it with the baseline which is only dependent on dialogue context.

7.3.2 Structured DST with domain reasoning

Implement and develop the idea of graph-based DST transferring knowledge and modeling relations of slot, value, and knowledge nodes. Investigate how different components are organized and performed in task-oriented dialogues. The knowledge can be common-sense relations or domain-specific for corresponding slots, which requires further studies.

The graph-based method is excelled in sharing common knowledge across multi-domains as well as incorporating external knowledge bases, paving the way for us to understand what dialogue systems have actually learned about user goals, and what depth of language understanding is needed to accomplish desired tasks.

7.3.3 Proposing a task for task-oriented dialogue simulation

Evaluate a dialogue system via its accomplishment of user goals and its efficiency in delivering information to help users. Simulate variable and comprehensive utterances to form multi-turn dialogues in broader domains. With the knowledge transfer in task-oriented dialogues being investigated and dialogue flows being modeled, we can evaluate a task-oriented dialogue via its accomplishment of user goals and efficiency based on whether the information related to identifying user goals is explicit in the dialogue. Simulate more variable and comprehensive utterances to form multi-turn dialogues based on our understanding of dialogue states in order to reduce the effort of deploying the dialogue system on broader domains, which is still elusive in the NLP community and will be useful for future research.

Appendices

APPENDIX A

EXPERIMENTAL DETAILS

A.1 Experimental Details for CoFunDST

A.1.1 Datasets

The domain-agnostic QA dataset combines six extractive QA datasets [229, 230, 231, 232, 233, 234] and two multi-choice QA datasets [179, 235]. We randomly sample 20% of examples in it to save computational time. Furthermore, negative sampling [151] is also adopted for this dataset. MultiWOZ 2.1 [159] is a cross-domain task-oriented dialogue dataset collected from 7 domains containing over 10,000 multi-turn dialogues. Our evaluation is conducted on the 5 most frequent domains, *i.e.* $\{restaurant, hotel, train, attraction, taxi\}$, in MultiWOZ 2.1 and follows dataset setups in [58].

The statistics of the QA training dataset are shown in Table Table A.1.

A.1.2 Baselines

TRADE [58] is an encoder-decoder model which leverages slot gates and copy mechanism and shares parameters for predicting unseen slot-values. SUMBT [161] proposes to use pre-trained BERT [103] to learn the relations between slot-types and values appearing in utterances and predicts dialogue states with slot-utterance matching. MA-DST [162] encodes dialogue context and domain-slots with attention mechanisms at multiple granularities to enhance the robustness of DST. TransferQA [151] is similar to ours which is

Table A.1: Statistics of the domain-agnostic question-answering (QA) dataset.

QA Dataset	<i>#Extractive</i>	<i>#Multiple-Choice</i>
<i>train</i>	566,262	93,982
<i>dev</i>	58,221	13,902

based on domain-agnostic QA training, while we extend this idea with knowledge fusion and directly use values in domain ontology as candidate choices to conform predictions of DST.

A.1.3 Model and Training Details

The number of parameters for models based on T5-Small [155] is 60 million. The 20% of combined domain-agnostic QA data is trained 6 epochs for 6.5h on NVIDIA RTX 3080 drivers, with the batch size 256. The average time for reference on MultiWoz 2.1 is 10 minutes.

A.1.4 Validation Results

Table A.2: The slot goal accuracy of non-categorical and categorical slots on MultiWoz 2.1, respectively. *#non-cat* and *#cat* are the total numbers of non-categorical and categorical slots for dev.

Domain	<i>#non-cat</i>	<i>#cat</i>	<i>Non-Categorical</i>	<i>Categorical</i>	<i>All</i>
Hotel	4911	7609	66.91	55.83	68.05
Train	8832	2514	61.62	39.24	68.12
Restaurant	5800	4967	60.14	45.69	62.27
Attraction	1249	3253	51.61	35.38	52.94
Taxi	1753	0	75.55	-	75.55

A.1.5 Error Analyses

Table A.3: Typical errors generated by the model on MultiWoz 2.1 dev set.

Error Type	Dialogue Context	Ground-Truth Slot-Values	Generated Slot-Values
false negative	USER: Hi. I will be attending a conference in the eastern part of town and wondered if you could help me find an available room.	hotel-area-east	
	SYSTEM: There are several guesthouses and 1 hotel in the east part of town. What price range are you looking for?	hotel-pricerange-expensive	hotel-type-guesthouse
	USER: I prefer a guesthouse. Do you have any expensive ones?	hotel-type-guesthouse	
false positive	USER: I am visiting cambridge and would like some suggestions for an upscale restaurant which serves modern global cuisine.	restaurant-food-chinese	restaurant-food-chinese
	SYSTEM: I am sorry. There is no matching restaurant. Would you like to try a different food type or price range?	restaurant-pricerange-expensive	restaurant-pricerange-expensive
	USER: Yes, how about a chinese restaurant?		restaurant-area-modern-global-cuisine
wrong prediction	USER: Can you recommend some fun entertainment in the centre?	attraction-type-boat	attraction-type-boat
	SYSTEM: I have many options available. What type of attraction do you prefer?	attraction-area-centre	attraction-area-boat
	USER: I prefer something related to boats.		

APPENDIX B

DATASETS

B.1 Statistics of Datasets

Table B.1: Statistics of SGD and MWOZ 2.1.

Dataset	Domains	<i>Train</i>	<i>Dev</i>	<i>Test</i>	<i>Train samples</i>	<i>Dev samples</i>	<i>Test samples</i>	Avg. Turns
SGD	18	5,278	761	1,531	38,745	5,589	11,349	14.7
MultiWOZ 2.1	5	8,324	999	1,000	12,426	987	1,091	13.5

Table B.2: Statistics of datasets for CLICKER.

Dataset	Mickey Corpus	X-CSQA	X-CODAH
<i>Train</i>	16K	8K (EN)	8K (EN)
<i>Dev</i>	1,000	500	400
<i>Test</i>	1,000	500	400

B.2 Service Glossary in Schema-Guided-Dialogue

Table B.3: Service names and corresponding domains in Schema-Guided-Dialogue.

SGD Services (Training samples)				
homes_1 (1829), hotels_4 (559), trains_1 (415), flights_2(822), movies_1 (1873), buses_1 (1054), music_2 (857), media_1 (1207), calendar_1 (773),	hotels_1 (868), music_1 (468), flights_3 (420), movies_2(118), rentalcars_1 (840), services_1 (1241), movies_3 (231), payment_1 (233), banks_1 (1138),	rentalcars_3 (332), restaurants_1 (2098), services_3 (959), music_3 (112), ridesharing_2 (380), buses_3 (405), flights_4 (290), services_2 (917), events_3 (220),	hotels_3 (737), rentalcars_2 (631), flights_1 (4680), media_3 (327), hotels_2 (1569), alarm_1 (367), events_1 (1424), hotels_1 (1593), banks_2 (312)	media_2 (215), homes_2 (424), services_4 (680), ridesharing_1 (412), restaurants_2 (807), events_2 (3537), weather_1 (259), hotels_1 (112),

APPENDIX C
ANALYSES

C.1 Statistical Significance of Reported Scores for SMETOD

Table C.1: Mean and standard deviation of all reported scores in Table Table 5.8 and Table Table 5.9 using 5 randomly sampled batch sizes, which are the same for all models and datasets. Student paired t-test shows $p < 0.01$ for scores changed by batch size. Combined = (Inform + Success) \times 0.5 + BLEU.

Model	Dataset	Module	Metric	Mean	Std
T5_{small}	MultiWOZ 2.1	NLU	JGA	59.69	0.028
		NLG	Inform	92.50	0.167
			Sucess	74.00	0.335
			BLEU	16.89	0.019
	Combined		100.14	-	
	MultiWOZ 2.2	NLU	JGA	59.60	0.026
		NLG	Inform	89.6	0.207
			Sucess	76.2	0.349
BLEU			17.1	0.031	
Combined	100.1		-		
T5_{base}	MultiWOZ 2.1	NLU	JGA	60.36	0.017
		NLG	Inform	92.3	0.071
			Sucess	78.8	0.217
			BLEU	16.88	0.011
	Combined		102.43	-	
	MultiWOZ 2.2	NLU	JGA	60.08	0.026
		NLG	Inform	89.0	0.182
			Sucess	76.0	0.349
BLEU			17.6	0.013	
Combined	99.7		-		

C.2 Ablation Study for SMETOD

Table C.2: Ablation on the positions of SoftMoE layer in T5.

Placement	JGA	F1	Precision	Recall
Encoder; Replace the first linear layer	61.98	93.35	0.94	0.93
Encoder; Replace the second linear layer	63.10	93.73	0.94	0.94
Encoder; Between first and second linear layer	62.68	93.58	0.94	0.93

REFERENCES

- [1] B. Azvine, D. Djian, K. C. Tsui, and W. Wobcke, “The intelligent assistant: An overview,” *Intelligent systems and soft computing: Prospects, tools and applications*, pp. 215–238, 2000.
- [2] S. Mallios and N. Bourbakis, “A survey on human machine dialogue systems,” in *2016 7th international conference on information, intelligence, systems & applications (iisa)*, IEEE, 2016, pp. 1–7.
- [3] Z. Tiancheng, “Learning to converse with latent actions,” Ph.D. dissertation, PhD dissertation, Carnegie Mellon University. 25, 27, 35, 2019.
- [4] C.-W. Huang and Y.-N. Chen, “Learning asr-robust contextualized embeddings for spoken language understanding,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 8009–8013.
- [5] Y. Weng *et al.*, “Joint contextual modeling for asr correction and language understanding,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 6349–6353.
- [6] F.-H. Yu, K.-Y. Chen, and K.-H. Lu, “Non-autoregressive asr modeling using pre-trained language models for chinese speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1474–1482, 2022.
- [7] P. A. Heeman, D. Byron, and J. F. Allen, “Identifying discourse markers in spoken dialog,” *Discourse*, vol. 58298, no. 6163, p. 8278, 1998.
- [8] Z. Zhang, T. Guo, and M. Chen, “Dialoguebert: A self-supervised learning based dialogue pre-training encoder,” in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3647–3651.
- [9] J. Cheng *et al.*, “Conversational semantic parsing for dialog state tracking,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 8107–8117.
- [10] J. Perez and F. Liu, “Dialog state tracking, a machine reading approach using memory network,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, M. Lapata, P. Blunsom, and A. Koller, Eds., Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 305–314.

- [11] C.-S. Wu, S. C. Hoi, R. Socher, and C. Xiong, “TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 917–929.
- [12] P. Budzianowski and I. Vulić, “Hello, it’s gpt-2—how can i help you? towards the use of pretrained language models for task-oriented dialogue systems,” *arXiv preprint arXiv:1907.05774*, 2019.
- [13] E. Hosseini-Asl, B. McCann, C.-S. Wu, S. Yavuz, and R. Socher, “A simple language model for task-oriented dialogue,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 179–20 191, 2020.
- [14] L. Cao, “Diaggpt: An llm-based chatbot with automatic topic management for task-oriented dialogue,” *arXiv preprint arXiv:2308.08043*, 2023.
- [15] D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai, “A form-based dialogue manager for spoken language applications,” in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP ’96*, vol. 2, 1996, 701–704 vol.2.
- [16] M. Eric, L. Krishnan, F. Charette, and C. D. Manning, “Key-value retrieval networks for task-oriented dialogue,” in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, Saarbrücken, Germany: Association for Computational Linguistics, Aug. 2017, pp. 37–49.
- [17] L. Qin, X. Xu, W. Che, Y. Zhang, and T. Liu, “Dynamic fusion network for multi-domain end-to-end task-oriented dialog,” *arXiv preprint arXiv:2004.11019*, 2020.
- [18] T.-H. Wen *et al.*, “A network-based end-to-end trainable task-oriented dialogue system,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Association for Computational Linguistics, 2017, pp. 438–449.
- [19] D. Thulke, N. Daheim, C. Dugast, and H. Ney, “Efficient retrieval augmented generation from unstructured knowledge for task-oriented dialog,” *arXiv preprint arXiv:2102.04643*, 2021.
- [20] W. Shen, Y. Gao, C. Huang, F. Wan, X. Quan, and W. Bi, “Retrieval-generation alignment for end-to-end task-oriented dialogue system,” *arXiv preprint arXiv:2310.08877*, 2023.
- [21] A. Dubey *et al.*, “The llama 3 herd of models,” *arXiv preprint arXiv:2407.21783*, 2024.

- [22] H. Touvron *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [24] T. Brown *et al.*, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901.
- [25] OpenAI *et al.*, *Gpt-4 technical report*, 2024. arXiv: 2303.08774 [cs.CL].
- [26] M. Chen *et al.*, “Evaluating large language models trained on code,” *arXiv preprint arXiv:2107.03374*, 2021.
- [27] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, “Language models as zero-shot planners: Extracting actionable knowledge for embodied agents,” in *International conference on machine learning*, PMLR, 2022, pp. 9118–9147.
- [28] R. Zellers *et al.*, “Defending against neural fake news,” *Advances in neural information processing systems*, vol. 32, 2019.
- [29] Q. Liu, N. Chen, T. Sakai, and X.-M. Wu, “A first look at llm-powered generative news recommendation,” *CoRR*, 2023.
- [30] K.-L. Chiu, A. Collins, and R. Alexander, “Detecting hate speech with gpt-3,” *arXiv preprint arXiv:2103.12407*, 2021.
- [31] V. Balaraman, S. Sheikhalishahi, and B. Magnini, “Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey,” in *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, H. Li *et al.*, Eds., Singapore and Online: Association for Computational Linguistics, Jul. 2021, pp. 239–251.
- [32] D. Yang *et al.*, “Refgpt: Dialogue generation of gpt, by gpt, and for gpt,” *arXiv preprint arXiv:2305.14994*, 2023.
- [33] I. Sekulić *et al.*, “Reliable llm-based user simulator for task-oriented dialogue systems,” *arXiv preprint arXiv:2402.13374*, 2024.
- [34] C. Niu, X. Wang, X. Cheng, J. Song, and T. Zhang, “Enhancing dialogue state tracking models through llm-backed user-agents simulation,” *arXiv preprint arXiv:2405.13037*, 2024.

- [35] R. Su, J. Yang, T.-W. Wu, and B.-H. Juang, “Choice fusion as knowledge for zero-shot dialogue state tracking,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [36] R. Su, T.-W. Wu, and B.-H. Juang, “Schema graph-guided prompt for multi-domain dialogue state tracking,” *arXiv preprint arXiv:2311.06345*, 2023.
- [37] R. Su, T.-W. Wu, and B.-H. Juang, “Act-aware slot-value predicting in multi-domain dialogue state tracking,” in *Interspeech 2021*, 2021, pp. 236–240.
- [38] R. Su and B.-H. Juang, “Many hands make light work: Task-oriented dialogue system with module-based mixture-of-experts,” *arXiv preprint arXiv:2405.09744*, 2024.
- [39] R. Su, Z. Sun, S. Lu, C. Ma, and C. Guo, “Clicker: Attention-based cross-lingual commonsense knowledge transfer,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [40] B. Grosz, D. Scott, H. Kamp, P. Cohen, and E. Giachin, *Discourse and dialogue*, 1995.
- [41] B. J. Grosz, C. Pollack, and C. Sidner, “Computational models of discourse,” 1989.
- [42] J. R. Hobbs, M. E. Stickel, D. E. Appelt, and P. Martin, “Interpretation as abduction,” *Artificial intelligence*, vol. 63, no. 1-2, pp. 69–142, 1993.
- [43] J. R. Searle, “Collective intentions and actions,” *Intentions in communication*, vol. 401, 1990.
- [44] J. R. Hobbs, “On the coherence and structure of discourse,” 1985.
- [45] B. J. Grosz and C. L. Sidner, “Attention, intentions, and the structure of discourse,” *Computational linguistics*, vol. 12, no. 3, pp. 175–204, 1986.
- [46] W. C. Mann and S. A. Thompson, *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles, 1987.
- [47] P. Holmes-Higgin, “Text generation—using discourse strategies and focus constraints to generate natural language text by kathleen r. mckeown, cambridge university press, 1992, pp 246, £13.95, isbn 0-521-43802-0.” *The Knowledge Engineering Review*, vol. 9, no. 4, pp. 421–422, 1994.
- [48] D. W. Angel, “The four types of conversations: Debate, dialogue, discourse, and diatribe,” *The Opportune Conflict*, 2016.

- [49] D. Walton and E. C. Krabbe, *Commitment in dialogue: Basic concepts of interpersonal reasoning*. State University of New York Press, 1995.
- [50] G. Ferguson and J. Allen, “Mixed-initiative systems for collaborative problem solving,” *AI magazine*, vol. 28, no. 2, pp. 23–23, 2007.
- [51] J. E. Allen, C. I. Guinn, and E. Horvitz, “Mixed-initiative interaction,” *IEEE Intelligent Systems and their Applications*, vol. 14, no. 5, pp. 14–23, 1999.
- [52] P. R. Cohen and H. J. Levesque, “Intention is choice with commitment,” *Artificial Intelligence*, vol. 42, no. 2, pp. 213–261, 1990.
- [53] D. R. Traum, *Foundations of rational agency, chapter speech acts for dialogue agents*, 1999.
- [54] I. Mporas, D. P. Lyras, K. N. Sgarbas, and N. Fakotakis, “Detection of dialogue acts using perplexity-based word clustering,” in *Text, Speech and Dialogue: 10th International Conference, TSD 2007, Pilsen, Czech Republic, September 3-7, 2007. Proceedings 10*, Springer, 2007, pp. 638–643.
- [55] N. Reithinger, R. Engel, M. Kipp, and M. Klesen, “Predicting dialogue acts for a speech-to-speech translation system,” in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP’96, IEEE*, vol. 2, 1996, pp. 654–657.
- [56] D. Bohus and A. Rudnicky, “A “k hypotheses+ other” belief updating model,” 2006.
- [57] S. Young, M. Gašić, B. Thomson, and J. D. Williams, “Pomdp-based statistical spoken dialog systems: A review,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.
- [58] C.-S. Wu, A. Madotto, E. Hosseini-Asl, C. Xiong, R. Socher, and P. Fung, “Transferable multi-domain state generator for task-oriented dialogue systems,” *arXiv preprint arXiv:1905.08743*, 2019.
- [59] V. Zhong, C. Xiong, and R. Socher, “Global-locally self-attentive dialogue state tracker,” *arXiv preprint arXiv:1805.09655*, 2018.
- [60] A. Rastogi, D. Hakkani-Tur, and L. Heck, “Scalable multi-domain dialogue state tracking,” in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Okinawa, Japan: IEEE, Dec. 2017.

- [61] H. Shi, T. Ushio, M. Endo, K. Yamagami, and N. Horii, “A multichannel convolutional neural network for cross-language dialog state tracking,” in *2016 IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 559–564.
- [62] L. Zhou and K. Small, “Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering,” *arXiv preprint arXiv:1911.06192*, 2019.
- [63] J.-G. Zhang *et al.*, “Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking,” *arXiv preprint arXiv:1910.03544*, 2019.
- [64] S. Gao, A. Sethi, S. Agarwal, T. Chung, and D. Hakkani-Tur, “Dialog state tracking: A neural reading comprehension approach,” *arXiv preprint arXiv:1908.01946*, 2019.
- [65] S. Gao, S. Agarwal, T. Chung, D. Jin, and D. Hakkani-Tur, “From machine reading comprehension to dialogue state tracking: Bridging the gap,” *arXiv preprint arXiv:2004.05827*, 2020.
- [66] A. Chotimongkol, “Learning the structure of task-oriented conversations from the corpus of in-domain dialogs,” Ph.D. dissertation, Carnegie Mellon University, Language Technologies Institute, School of . . . , 2008.
- [67] M. F. McTear, “Spoken dialogue technology: Enabling the conversational user interface,” *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 90–169, 2002.
- [68] J. Weizenbaum, “Eliza—a computer program for the study of natural language communication between man and machine,” *Commun. ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966.
- [69] K. M. Colby, F. D. Hilf, S. Weber, and H. C. Kraemer, “Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes,” *Artificial Intelligence*, vol. 3, pp. 199–221, 1972.
- [70] R. S. Wallace, *The anatomy of ALICE*. Springer, 2009.
- [71] R. Carpenter, *Cleverbot*. Mar. 2017.
- [72] L. Zhou, J. Gao, D. Li, and H.-Y. Shum, “The design and implementation of xiaoice, an empathetic social chatbot,” *Computational Linguistics*, vol. 46, no. 1, pp. 53–93, 2020.
- [73] A. Schmitt, S. Ultes, and W. Minker, “A parameterized and annotated spoken dialog corpus of the cmu let’s go bus information system.,” in *LREC*, 2012, pp. 3369–3373.

- [74] A. Bordes, Y.-L. Boureau, and J. Weston, “Learning end-to-end goal-oriented dialog,” *arXiv preprint arXiv:1605.07683*, 2016.
- [75] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd, “Gus, a frame-driven dialog system,” *Artificial intelligence*, vol. 8, no. 2, pp. 155–173, 1977.
- [76] P. R. Cohen, A. Cheyer, M. Wang, and S. C. Baeg, “An open agent architecture,” in *AAAI Spring Symposium*, vol. 1, 1994.
- [77] D. Litman and S. Silliman, “Itspoke: An intelligent tutoring spoken dialogue system,” in *Demonstration papers at HLT-NAACL 2004*, 2004, pp. 5–8.
- [78] T. Fu, S. Gao, X. Zhao, J.-r. Wen, and R. Yan, “Learning towards conversational ai: A survey,” *AI Open*, vol. 3, pp. 14–28, 2022.
- [79] J. J. Godfrey, E. C. Holliman, and J. McDaniel, “Switchboard: Telephone speech corpus for research and development,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, IEEE Computer Society, vol. 1, 1992, pp. 517–520.
- [80] P. Kingsbury, S. Strassel, C. McLemore, and R. McIntyre, “Callhome american english transcripts, ldc97t14,” *Philadelphia: Linguistic Data Consortium*, 1997.
- [81] N. Mrkšić, D. O. Séaghdha, T.-H. Wen, B. Thomson, and S. Young, “Neural belief tracker: Data-driven dialogue state tracking,” *arXiv preprint arXiv:1606.03777*, 2016.
- [82] A. Rastogi, X. Zang, S. Sunkara, R. Gupta, and P. Khaitan, “Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 8689–8696.
- [83] B. Liu and I. Lane, “An end-to-end trainable neural network model with belief tracking for task-oriented dialog,” *arXiv preprint arXiv:1708.05956*, 2017.
- [84] A. Rastogi, D. Hakkani-Tür, and L. Heck, “Scalable multi-domain dialogue state tracking,” in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2017, pp. 561–568.
- [85] H. Shi, T. Ushio, M. Endo, K. Yamagami, and N. Horii, “A multichannel convolutional neural network for cross-language dialog state tracking,” in *2016 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2016, pp. 559–564.

- [86] V. Zhong, C. Xiong, and R. Socher, “Global-locally self-attentive dialogue state tracker,” *arXiv preprint arXiv:1805.09655*, 2018.
- [87] L. Chen, B. Lv, C. Wang, S. Zhu, B. Tan, and K. Yu, “Schema-guided multi-domain dialogue state tracking with graph attention neural networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 7521–7528.
- [88] L. Liao, Y. Ma, W. Lei, and T.-S. Chua, “Rethinking dialogue state tracking with reasoning,” *arXiv preprint arXiv:2005.13129*, 2020.
- [89] S. Kim, S. Yang, G. Kim, and S.-W. Lee, “Efficient dialogue state tracking by selectively overwriting memory,” *arXiv preprint arXiv:1911.03906*, 2019.
- [90] M. Heck *et al.*, “Trippy: A triple copy strategy for value independent neural dialog state tracking,” *arXiv preprint arXiv:2005.02877*, 2020.
- [91] L. Ren, J. Ni, and J. McAuley, “Scalable and accurate dialogue state tracking via hierarchical sequence generation,” *arXiv preprint arXiv:1909.00754*, 2019.
- [92] H. Le, R. Socher, and S. C. Hoi, “Non-autoregressive dialog state tracking,” in *International Conference on Learning Representations*, 2020.
- [93] V. Balaraman and B. Magnini, “Domain-aware dialogue state tracker for multi-domain dialogue systems,” *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 29, pp. 866–873, Jan. 2021.
- [94] V. Noroozi, Y. Zhang, E. Bakhturina, and T. Kornuta, “A fast and robust bert-based dialogue state tracker for schema-guided dialogue dataset,” *arXiv preprint arXiv:2008.12335*, 2020.
- [95] M. D. Sadek, P. Bretier, and F. Panaget, “Artimis: Natural dialogue meets rational agency,” *IJCAI (2)*, vol. 1030, p. 1035, 1997.
- [96] R. Freedman, “Plan-based dialogue management in a physics tutor,” in *Sixth Applied Natural Language Processing Conference*, 2000, pp. 52–59.
- [97] B. Thomson and S. Young, “Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems,” *Computer Speech & Language*, vol. 24, no. 4, pp. 562–588, 2010.
- [98] P.-H. Su *et al.*, “Continuously learning neural dialogue management,” *arXiv preprint arXiv:1606.02689*, 2016.
- [99] N. Bang, J. Lee, and M.-W. Koo, “Task-optimized adapters for an end-to-end task-oriented dialogue system,” *arXiv preprint arXiv:2305.02468*, 2023.

- [100] Y. Zhang *et al.*, “DIALOGPT : Large-scale generative pre-training for conversational response generation,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Association for Computational Linguistics, 2020, pp. 270–278.
- [101] B. Peng, C. Li, J. Li, S. Shayandeh, L. Liden, and J. Gao, “Soloist: Building task bots at scale with transfer learning and machine teaching,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 807–824, 2021.
- [102] W. He *et al.*, “Unified dialog model pre-training for task-oriented dialog understanding and generation,” in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 187–200.
- [103] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [104] L. Dong *et al.*, “Unified language model pre-training for natural language understanding and generation,” *Advances in neural information processing systems*, vol. 32, 2019.
- [105] C. Raffel *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.
- [106] N. Houlsby *et al.*, “Parameter-efficient transfer learning for nlp,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 2790–2799.
- [107] X. L. Li and P. Liang, “Prefix-tuning: Optimizing continuous prompts for generation,” *arXiv preprint arXiv:2101.00190*, 2021.
- [108] B. Lester, R. Al-Rfou, and N. Constant, “The power of scale for parameter-efficient prompt tuning,” *arXiv preprint arXiv:2104.08691*, 2021.
- [109] Q. Wang *et al.*, “Divide, conquer, and combine: Mixture of semantic-independent experts for zero-shot dialogue state tracking,” *arXiv preprint arXiv:2306.00434*, 2023.
- [110] A. Rücklé *et al.*, “Adapterdrop: On the efficiency of adapters in transformers,” *arXiv preprint arXiv:2010.11918*, 2020.
- [111] Y. Feng, A. Lipani, F. Ye, Q. Zhang, and E. Yilmaz, “Dynamic schema graph fusion network for multi-domain dialogue state tracking,” *arXiv preprint arXiv:2204.06677*, 2022.

- [112] C.-H. Lee, H. Cheng, and M. Ostendorf, “Dialogue state tracking with a language model using schema-driven prompting,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4937–4949.
- [113] Y. Yang, W. Lei, J. Cao, J. Li, and T.-S. Chua, “Prompt learning for few-shot dialogue state tracking,” *arXiv preprint arXiv:2201.05780*, 2022.
- [114] Q. Wang, Y. Cao, P. Li, Y. Fu, Z. Lin, and L. Guo, “Slot dependency modeling for zero-shot cross-domain dialogue state tracking,” in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 510–520.
- [115] Y. Hu, C.-H. Lee, T. Xie, T. Yu, N. A. Smith, and M. Ostendorf, “In-context learning for few-shot dialogue state tracking,” *arXiv preprint arXiv:2203.08568*, 2022.
- [116] R. Gupta, H. Lee, J. Zhao, A. Rastogi, Y. Cao, and Y. Wu, “Show, don’t tell: Demonstrations outperform descriptions for schema-guided task-oriented dialogue,” *arXiv preprint arXiv:2204.04327*, 2022.
- [117] P. Venkateswaran, E. Duesterwald, and V. Isahagian, “District: Dialogue state tracking with retriever driven in-context tuning,” *arXiv preprint arXiv:2212.02851*, 2022.
- [118] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [119] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.
- [120] Y. Liu *et al.*, “Multilingual denoising pre-training for neural machine translation,” *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020.
- [121] L. Xue *et al.*, “MT5: A massively multilingual pre-trained text-to-text transformer,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Linguistics, Jun. 2021, pp. 483–498.
- [122] J.-K. Kim, Y.-B. Kim, R. Sarikaya, and E. Fosler-Lussier, “Cross-lingual transfer learning for POS tagging without cross-lingual resources,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2832–2838.

- [123] W. Ahmad, Z. Zhang, X. Ma, E. Hovy, K.-W. Chang, and N. Peng, “On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 2440–2452.
- [124] J. Xie, Z. Yang, G. Neubig, N. A. Smith, and J. Carbonell, “Neural cross-lingual named entity recognition with minimal resources,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 369–379.
- [125] A. Conneau *et al.*, “Unsupervised cross-lingual representation learning at scale,” *CoRR*, vol. abs/1911.02116, 2019. arXiv: 1911.02116.
- [126] S. Wu and M. Dredze, “Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT,” *CoRR*, vol. abs/1904.09077, 2019. arXiv: 1904.09077.
- [127] T. Pires, E. Schlinger, and D. Garrette, “How multilingual is multilingual BERT?” In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4996–5001.
- [128] A. Siddhant *et al.*, “Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation,” *CoRR*, vol. abs/1909.00437, 2019. arXiv: 1909.00437.
- [129] V. Hudeček and O. Dušek, “Are llms all you need for task-oriented dialogue?” *arXiv preprint arXiv:2304.06556*, 2023.
- [130] D. Ghosal, N. Majumder, S. Poria, N. Chhaya, and A. Gelbukh, “Dialoguecn: A graph convolutional neural network for emotion recognition in conversation,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 154–164.
- [131] L. Chen, B. Tan, S. Long, and K. Yu, “Structured dialogue policy with graph neural networks,” in *COLING*, 2018.
- [132] L. Qin, W. Che, Y. Li, M. Ni, and T. Liu, “Dcr-net: A deep co-interactive relation network for joint dialog act recognition and sentiment classification,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 8665–8672.

- [133] L. Qin, Z. Li, W. Che, M. Ni, and T. Liu, “Co-gat: A co-interactive graph attention network for joint dialog act recognition and sentiment classification,” *arXiv preprint arXiv:2012.13260*, 2020.
- [134] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *stat*, vol. 1050, p. 20, 2017.
- [135] J. Guo, K. Shuang, J. Li, Z. Wang, and Y. Liu, “Beyond the granularity: Multi-perspective dialogue collaborative selection for dialogue state tracking,” *arXiv preprint arXiv:2205.10059*, 2022.
- [136] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, “Adapterfusion: Non-destructive task composition for transfer learning,” *arXiv preprint arXiv:2005.00247*, 2020.
- [137] R. Karimi Mahabadi, J. Henderson, and S. Ruder, “Compacter: Efficient low-rank hypercomplex adapter layers,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 1022–1035, 2021.
- [138] Y. Gu, X. Han, Z. Liu, and M. Huang, “Ppt: Pre-trained prompt tuning for few-shot learning,” *arXiv preprint arXiv:2109.04332*, 2021.
- [139] T. Vu, B. Lester, N. Constant, R. Al-Rfou, and D. Cer, “Spot: Better frozen model adaptation through soft prompt transfer,” *arXiv preprint arXiv:2110.07904*, 2021.
- [140] A. Madotto *et al.*, “Continual learning in task-oriented dialogue systems,” *arXiv preprint arXiv:2012.15504*, 2020.
- [141] Q. Zhu, B. Li, F. Mi, X. Zhu, and M. Huang, “Continual prompt tuning for dialog state tracking,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2022.
- [142] Y. Su *et al.*, “Multi-task pre-training for plug-and-play task-oriented dialogue system,” 2022.
- [143] Y. Mo, J. Yoo, and S. Kang, “Parameter-efficient fine-tuning method for task-oriented dialogue systems,” *Mathematics*, vol. 11, no. 14, p. 3048, 2023.
- [144] T. Brown *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [145] D. Lepikhin *et al.*, “Gshard: Scaling giant models with conditional computation and automatic sharding,” *arXiv preprint arXiv:2006.16668*, 2020.

- [146] W. Fedus, B. Zoph, and N. Shazeer, “Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity,” *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 5232–5270, 2022.
- [147] N. Du *et al.*, “Glam: Efficient scaling of language models with mixture-of-experts,” in *International Conference on Machine Learning*, PMLR, 2022, pp. 5547–5569.
- [148] Y. Zhou *et al.*, “Mixture-of-experts with expert choice routing,” in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022.
- [149] P. Langley, B. Meadows, A. Gabaldon, and R. Heald, “Abductive understanding of dialogues about joint activities,” *Interaction Studies*, vol. 15, no. 3, pp. 426–454, 2014.
- [150] G. Campagna, A. Foryciarz, M. Moradshahi, and M. S. Lam, “Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking,” *arXiv preprint arXiv:2005.00891*, 2020.
- [151] Z. Lin *et al.*, “Zero-shot dialogue state tracking via cross-task transfer,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 7890–7900.
- [152] Y.-P. Ruan, Z.-H. Ling, J.-C. Gu, and Q. Liu, “Fine-tuning bert for schema-guided zero-shot dialogue state tracking,” *arXiv preprint arXiv:2002.00181*, 2020.
- [153] P. Knoeferle, “Predicting (variability of) context effects in language comprehension,” *Journal of Cultural Cognitive Science*, vol. 3, no. 2, pp. 141–158, 2019.
- [154] S. Li *et al.*, “Zero-shot generalization in dialog state tracking through generative question answering,” *arXiv preprint arXiv:2101.08333*, 2021.
- [155] C. Raffel *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.
- [156] L. Hirschman, M. Light, E. Breck, and J. D. Burger, “Deep read: A reading comprehension system,” in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, 1999, pp. 325–332.
- [157] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392.

- [158] S. Wu, Y. Li, D. Zhang, Y. Zhou, and Z. Wu, “Diverse and informative dialogue generation with context-specific commonsense knowledge awareness,” in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 5811–5820.
- [159] M. Eric *et al.*, “Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines,” *arXiv preprint arXiv:1907.01669*, 2019.
- [160] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [161] H. Lee, J. Lee, and T.-Y. Kim, “Sumbt: Slot-utterance matching for universal and scalable belief tracking,” in *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 2019, pp. 5478–5483.
- [162] A. Kumar, P. Ku, A. Goyal, A. Metallinou, and D. Hakkani-Tur, “Ma-dst: Multi-attention-based scalable dialog state tracking,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 8107–8114.
- [163] N. Shazeer and M. Stern, “Adafactor: Adaptive learning rates with sublinear memory cost,” in *International Conference on Machine Learning*, PMLR, 2018, pp. 4596–4604.
- [164] Y. Feng, Y. Wang, and H. Li, “A sequence-to-sequence approach to dialogue state tracking,” *arXiv preprint arXiv:2011.09553*, 2020.
- [165] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.
- [166] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.
- [167] R. Joshi, V. Balachandran, S. Vashishth, A. Black, and Y. Tsvetkov, “Dialograph: Incorporating interpretable strategy-graph networks into negotiation dialogues,” in *International Conference on Learning Representations*, 2021.
- [168] E. Ranjan, S. Sanyal, and P. Talukdar, “Asap: Adaptive structure aware pooling for learning hierarchical graph representations,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 5470–5477.
- [169] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations*, 2019.

- [170] M. Heck *et al.*, “Robust dialogue state tracking with weak supervision and sparse data,” *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 1175–1192, 2022.
- [171] H. Zhang *et al.*, “Monet: Tackle state momentum via noise-enhanced training for dialogue state tracking,” *arXiv preprint arXiv:2211.05503*, 2022.
- [172] Y. Su *et al.*, “Multi-task pre-training for plug-and-play task-oriented dialogue system,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4661–4676.
- [173] S. Gao, R. Takanobu, A. Bosselut, and M. Huang, “End-to-end task-oriented dialog modeling with semi-structured knowledge management,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.
- [174] J. R. Searle and D. Vanderveken, “Speech acts and illocutionary logic,” in *Logic, Thought and Action*, D. Vanderveken, Ed. Dordrecht: Springer Netherlands, 2005, pp. 109–132, ISBN: 978-1-4020-3167-0.
- [175] N. Mrkšić, D. O. Séaghdha, T.-H. Wen, B. Thomson, and S. Young, “Neural belief tracker: Data-driven dialogue state tracking,” *arXiv preprint arXiv:1606.03777*, 2016.
- [176] A. Rastogi, X. Zang, S. Sunkara, R. Gupta, and P. Khaitan, “Schema-guided dialogue state tracking task at dstc8,” *arXiv preprint arXiv:2002.01359*, 2020.
- [177] K. Cho *et al.*, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [178] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” *arXiv preprint arXiv:1611.01603*, 2016.
- [179] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, “Race: Large-scale reading comprehension dataset from examinations,” *arXiv preprint arXiv:1704.04683*, 2017.
- [180] S. Kim, S. Yang, G. Kim, and S.-W. Lee, “Efficient dialogue state tracking by selectively overwriting memory,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 567–582.
- [181] Y. Shan *et al.*, “A contextual hierarchical attention network with adaptive objective for dialogue state tracking,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6322–6333.

- [182] J. Puigcerver, C. Riquelme, B. Mustafa, and N. Houlsby, “From sparse to soft mixtures of experts,” *arXiv preprint arXiv:2308.00951*, 2023.
- [183] Y. Bang, N. Lee, E. Ishii, A. Madotto, and P. Fung, “Assessing political prudence of open-domain chatbots,” in *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, H. Li *et al.*, Eds., Singapore and Online: Association for Computational Linguistics, Jul. 2021, pp. 548–555.
- [184] X. Zang, A. Rastogi, S. Sunkara, R. Gupta, J. Zhang, and J. Chen, “Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines,” in *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, ACL 2020*, 2020, pp. 109–117.
- [185] I. Casanueva, T. Temčinas, D. Gerz, M. Henderson, and I. Vulić, “Efficient intent detection with dual sentence encoders,” *arXiv preprint arXiv:2003.04807*, 2020.
- [186] S. Larson *et al.*, “An evaluation dataset for intent classification and out-of-scope prediction,” Association for Computational Linguistics, 2019, pp. 1311–1316.
- [187] X. Liu, A. Eshghi, P. Swietojanski, and V. Rieser, “Benchmarking natural language understanding services for building conversational agents,” *arXiv preprint arXiv:1903.05566*, 2019.
- [188] N. Shazeer *et al.*, “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer,” in *International Conference on Learning Representations*, 2017.
- [189] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [190] N. Shazeer *et al.*, “Mesh-tensorflow: Deep learning for supercomputers,” *Advances in neural information processing systems*, vol. 31, 2018.
- [191] A. Vaswani *et al.*, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [192] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015.
- [193] S. Mehri, M. Eric, and D. Hakkani-Tur, “Dialogue: A natural language understanding benchmark for task-oriented dialogue,” *arXiv preprint arXiv:2009.13570*, 2020.
- [194] W. He *et al.*, “SPACE-2: Tree-structured semi-supervised contrastive pre-training for task-oriented dialog understanding,” in *Proceedings of the 29th International*

Conference on Computational Linguistics, Gyeongju, Republic of Korea: International Committee on Computational Linguistics, 2022, pp. 553–569.

- [195] R. Cholakov and T. Kolev, “Efficient task-oriented dialogue systems with response selection as an auxiliary task,” *arXiv preprint arXiv:2208.07097*, 2022.
- [196] Y. Dai *et al.*, “Preview, attend and review: Schema-aware curriculum learning for multi-domain dialog state tracking,” *arXiv preprint arXiv:2106.00291*, 2021.
- [197] X. Tian *et al.*, “Amendable generation for dialogue state tracking,” *arXiv preprint arXiv:2110.15659*, 2021.
- [198] J. Zhao *et al.*, “Description-driven task-oriented dialog modeling,” *arXiv preprint arXiv:2201.08904*, 2022.
- [199] H. Jeon and G. G. Lee, “Domain state tracking for a simplified dialogue system,” *arXiv preprint arXiv:2103.06648*, 2021.
- [200] Q. Wu, J. Gung, R. Shu, and Y. Zhang, “Diacttod: Learning generalizable latent dialogue acts for controllable task-oriented dialogue systems,” *arXiv preprint arXiv:2308.00878*, 2023.
- [201] Y. Yang, Y. Li, and X. Quan, “Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 14 230–14 238.
- [202] Z. Lin, A. Madotto, G. I. Winata, and P. Fung, “Mintl: Minimalist transfer learning for task-oriented dialogue systems,” *arXiv preprint arXiv:2009.12005*, 2020.
- [203] Y. Feng *et al.*, “Fantastic rewards and how to tame them: A case study on reward learning for task-oriented dialogue systems,” *arXiv preprint arXiv:2302.10342*, 2023.
- [204] W. He *et al.*, “Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, 2022, pp. 10 749–10 757.
- [205] Y. Lee, “Improving end-to-end task-oriented dialog system with a simple auxiliary task,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 1296–1303.
- [206] X. Yu, Q. Wu, K. Qian, and Z. Yu, “Krls: Improving end-to-end response generation in task oriented dialog with reinforced keywords learning,” *arXiv preprint arXiv:2211.16773*, 2022.

- [207] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” *Advances in neural information processing systems*, vol. 12, 1999.
- [208] E. Davis and G. Marcus, “Commonsense reasoning and commonsense knowledge in artificial intelligence,” *Communications of the ACM*, vol. 58, no. 9, pp. 92–103, 2015.
- [209] A. Talmor, J. Herzig, N. Lourie, and J. Berant, “CommonsenseQA: A question answering challenge targeting commonsense knowledge,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4149–4158.
- [210] S. Storks, Q. Gao, and J. Y. Chai, “Recent advances in natural language inference: A survey of benchmarks, resources, and approaches,” *arXiv preprint arXiv:1904.01172*, 2019.
- [211] M. Roemmele, C. A. Bejan, and A. S. Gordon, “Choice of plausible alternatives: An evaluation of commonsense causal reasoning.,” in *AAAI spring symposium: logical formalizations of commonsense reasoning*, 2011, pp. 90–95.
- [212] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi, “WINOGRANDE: an adversarial winograd schema challenge at scale,” *CoRR*, vol. abs/1907.10641, 2019. arXiv: 1907.10641.
- [213] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, “SWAG: A large-scale adversarial dataset for grounded commonsense inference,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 93–104.
- [214] L. Huang, R. Le Bras, C. Bhagavatula, and Y. Choi, “Cosmos QA: Machine reading comprehension with contextual commonsense reasoning,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2391–2401.
- [215] D. Emelin and R. Sennrich, “Wino-X: Multilingual Winograd schemas for commonsense reasoning and coreference resolution,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 8517–8532.

- [216] D. Stojanovski, B. Kroyer, D. Peskov, and A. Fraser, “ContraCAT: Contrastive coreference analytical templates for machine translation,” in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 4732–4749.
- [217] E. M. Ponti, G. Glavas, O. Majewska, Q. Liu, I. Vulic, and A. Korhonen, “XCOPA: A multilingual dataset for causal commonsense reasoning,” *CoRR*, vol. abs/2005.00333, 2020. arXiv: 2005.00333.
- [218] B. Y. Lin, S. Lee, X. Qiao, and X. Ren, “Common sense beyond english: Evaluating and improving multilingual language models for commonsense reasoning,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP 2021)*, 2021.
- [219] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018. arXiv: 1810.04805.
- [220] G. Lample and A. Conneau, “Cross-lingual language model pretraining,” *arXiv preprint arXiv:1901.07291*, 2019.
- [221] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson, “XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization,” *CoRR*, vol. abs/2003.11080, 2020. arXiv: 2003.11080.
- [222] Z. Chi *et al.*, “InfoXLM: An information-theoretic framework for cross-lingual language model pre-training,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Linguistics, Jun. 2021, pp. 3576–3588.
- [223] A. Conneau *et al.*, “Xnli: Evaluating cross-lingual sentence representations,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, 2018.
- [224] J. Yeo, G. Wang, H. Cho, S. Choi, and S.-w. Hwang, “Machine-translated knowledge transfer for commonsense causal reasoning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.
- [225] S. Gururangan *et al.*, “Don’t stop pretraining: Adapt language models to domains and tasks,” *arXiv preprint arXiv:2004.10964*, 2020.
- [226] T. Wolf *et al.*, “Transformers: State-of-the-art natural language processing,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Pro-*

cessing: System Demonstrations, Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45.

- [227] Y. Wu *et al.*, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” *arXiv preprint arXiv:1609.08144*, 2016.
- [228] G. Team *et al.*, *Gemini: A family of highly capable multimodal models*, 2024. arXiv: 2312.11805 [cs.CL].
- [229] A. Trischler *et al.*, “Newsqa: A machine comprehension dataset,” *arXiv preprint arXiv:1611.09830*, 2016.
- [230] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, “Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension,” *arXiv preprint arXiv:1705.03551*, 2017.
- [231] M. Dunn, L. Sagun, M. Higgins, V. U. Guney, V. Cirik, and K. Cho, “Searchqa: A new q&a dataset augmented with context from a search engine,” *arXiv preprint arXiv:1704.05179*, 2017.
- [232] P. Rajpurkar, R. Jia, and P. Liang, “Know what you don’t know: Unanswerable questions for squad,” *arXiv preprint arXiv:1806.03822*, 2018.
- [233] Z. Yang *et al.*, “Hotpotqa: A dataset for diverse, explainable multi-hop question answering,” *arXiv preprint arXiv:1809.09600*, 2018.
- [234] T. Kwiatkowski *et al.*, “Natural questions: A benchmark for question answering research,” *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453–466, 2019.
- [235] K. Sun, D. Yu, J. Chen, D. Yu, Y. Choi, and C. Cardie, “Dream: A challenge data set and models for dialogue-based reading comprehension,” *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 217–231, 2019.

VITA

Ruolin Su is an Applied Scientist 2 for Microsoft's PowerPoint AI, where she developed innovative AI solutions for presentations and user interfaces, including a PowerPoint quality assessment tool powered by Large Language Models (LLMs). She pursues a PhD in Electrical and Computer Engineering at Georgia Institute of Technology, advised by Prof. Biing-Hwang Juang, with research interests in Natural Language Processing (NLP), Task-Oriented Dialogue (TOD) systems, and conversational service automation. Ruolin earned her B.S. in Communications Engineering with honors from Harbin Institute of Technology, China.

Her research on semantic representations in spoken language understanding (SLU), dialogue state tracking (DST) for automating dialogue services, and deep learning methods for sequential labeling tasks, aiming to automate task-oriented dialogue systems to facilitate various services via human-machine dialogues.

She has completed several impactful internships at leading tech companies, including Google, Amazon, and Interactions. At Google Display Ads, she worked on tuning-parameter exploration to enhance revenue-cost prediction and optimization, collaborating with Zibo Zhao, Nancy Fan, Ehsan Maani, and Sayali Kulkarni. At Amazon Alexa AI, she contributed to cross-lingual commonsense knowledge transfer, working alongside Zhongkai Sun, Cynthia Lu, Yunji Wang, and Chengyuan Ma. She also interned at Interactions, where she collaborated with John Chen on text generation techniques to improve natural language understanding for food service applications.