# Collaborative agents for task-oriented dialogue systems

Pei, J.

Dialogue systems (a.k.a. conversational agents) aim to help people interact with machines through natural language. They are playing an increasingly important role in our daily life.

There are two categories of approaches: modularized pipeline agents and end-to-end single-module agents. A challenge of the former is error accumulation because multiple modules are sequentially dependent. And concerning the latter, it is impractical to use a single general agent to handle all complex cases.

In this thesis, we introduce a new framework, namely collaborative task-oriented dialogue systems. Within this framework, we propose a series of approaches where a group of collaborative specialized agents outperforms a single general agent, in terms of four dimensions: (i) model collaboration, (ii) user collaboration, (iii) language collaboration, and (iv) uncertainty estimation.

ISBN 978-94-93278-33-2

COLLABORATIVE AGENTS FOR TASK-ORIENTED DIALOGUE SYSTEMS | JIAHUAN PEI

# COLLABORATIVE AGENTS FOR TASK-ORIENTED DIALOGUE SYSTEMS

JIAHUAN PEI

# Collaborative Agents for Task-oriented Dialogue Systems

**Jiahuan Pei**

# Collaborative Agents for Task-oriented Dialogue Systems

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in
de Agnietenkapel
op donderdag 22 december 2022, te 10:00 uur

door

Jiahuan Pei

geboren te Liaoning

**Promotiecommissie**

|  |  |  |
|---|---|---|
| Promotor: | prof. dr. M. de Rijke | Universiteit van Amsterdam |
| Co-promotor: | prof. dr. P. Ren | Shandong University |
| Overige leden: | prof. dr. C. Monz | Universiteit van Amsterdam |
|  | prof. dr. R. Fernández Rovira | Universiteit van Amsterdam |
|  | prof. dr. E. Kanoulas | Universiteit van Amsterdam |
|  | dr. M. Huang | Tsinghua University |
|  | prof. dr. E. Yilmaz | University College London |

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The research was supported by the China Scholarship Council under grant #201706060188.

Cover by Hao Xie
Printed by Off Page, Amsterdam

ISBN: 978-94-93278-33-2

## Acknowledgements

Pursuing a Ph.D. is a long and winding journey, and I am trying to answer Maarten's interview question: How do you face challenges? Luckily, I have made my best decision to join the IRLab and select Maarten de Rijke as my supervisor. I have never been alone on the way and finally almost arrived at my destination. Besides a deep dive into research, I am glad to see my personal growth and development in independence, maturity, calmness, and perseverance. I am sincerely grateful to my supervisors, friends, and family, who have always put up, supported, and helped me.

First, I would like to thank Prof. dr. Maarten de Rijke for being my supervisor and for his tremendous efforts during the whole Ph.D. journey. I am impressed by your rigorous attitude toward scientific research, your patience and passion for student supervision, and your kind and open-minded inspiration during meetings. We have had many discussions and explorations on many research topics. Thanks to your profound knowledge and insight into science, I have been able to work on an exciting and innovative research direction. You spent a lot of time and effort helping to make research plans, discuss ideas, and revise papers and this thesis. You supported me in attending academic conferences and seminars to develop myself and broaden my academic networks. During our talks about my career plan, you helped to open my mind and realize that someone who holds a Ph.D. is in a unique position to experience, and switch back and forth between, different roles on industrial and academic career paths. You are a highly esteemed mentor who is conscientious and dutiful in supervising students and has an outstanding academic reputation. Thank you so much, Maarten!

I want to thank Prof. dr. Pengjie Ren for being my co-supervisor and for his great efforts in helping me learn how to think and work as a researcher. You spent lots of time on daily supervision, including weekly meetings and discussions, and guided me on many specific research projects and by revising my papers and thesis. You are such a trustworthy and effective mentor who is kind, patient, and enlightening! Thanks to your talent and sharp sense of frontier research, the feasibility and innovative character of our ideas have been recognized and accepted in our academic field. You supported me in cooperating with students and peers and helped me plan my future career. Besides, I enjoyed our cooking and drinking times during traditional Chinese festivals. Thank you so much, Pengjie!

I am honored to have a Ph.D. committee of leading experts. Thank you, Prof. dr. Christof Monz, Prof. dr. Raquel Fernández Rovira, Prof. dr. Evangelos Kanoulas, Prof. dr. Minlie Huang, Prof. dr. Emine Yilmaz for your time and effort in reviewing my thesis and for your constructive feedback.

I would like to thank the China Scholarship Council Program for sponsoring my research under grant #201706060188.

Many people have helped me. I would like to thank Guojun Yan for his help with experiments. We had an excellent time working on two interesting academic papers. I would like to thank Hao Xie for designing the beautiful thesis cover. I would like to thank Dr. Dan Li and Spyretta Leivaiti for being my paranymphs for my doctoral defense. I would like to thank Petra and Pablo for the wonderful activities and organizations, especially Petra, for helping me with university procedures. I would like to thank all my colleagues from the IRLab at the University of Amsterdam. It has been an inspiring experience to have worked closely together with you. I have greatly enjoyed our research activities like Soos talks, Research meetings, and SEA, as well as our social events like coffee breaks and Friday drinks. Many thanks to Ali A, Ali V, Amir, Ana, Andrew, Anna, Antonios, Arezoo, Barrie, Chang, Chuan, Clemencia, Dan, David, Gabriel, Georgios, Hamid, Harrie, Hinda, Hongyu, Hosein, Ilya, Jie, Jin, Julien, Maarten M, Maartje, Maria, Mariya, Marzieh, Maurits, Masha, Ming, Mohammad, Mostafa, Mounia, Mozhdeh, Nikos, Olivier, Peilei, Pengjie, Philipp, Pooya, Praveen, Rolf, Romain, Ruben, Sam, Sami, Sebastian, Shaojie, Shashank, Shubha, Spyretta, Svitlana, Thilina, Thong, Vaishali, Vera, Wanyu, Weijia, Xiangsheng, Xiaohui, Xinyi, Yang, Yangjun, Yifan, Zhaochun, and Ziming. Special thanks to Maarten for helping me translate the Summary into the Samenvatting.

I was fortunate enough to enjoy internships and gain working experience at Amazon. I want to thank my mentors, Dr. Cheng Wang, Dr. Gyuri Szarvas, and Dr. Daniil Sorokin, for their kind and enlightening mentoring during my internships. I want to thank my manager, Dr. Omar Zaidan, for helping me relocate to Berlin and back to Amazon.

I would like to thank my friends who supported and accompanied me along the way: Amy, Benyou, Dan, Hanzhen, Hao, Hui, Jiayun, Kefan, Kim, Lihui, Mike, Qi, Shihan, Shuo, Spyretta, Wei, Wu, Xiaowei, Zeyuan, Zhe. I really enjoy our happy times together.

Last but not least, I would like to thank my family, my dear mother, father, grandfather, aunt, sisters, and my first love. Thank you for your love, encouragement, and support! That is what keeps me going!

Jiahuan Pei
August 2022
Amsterdam

# Contents

# 1
# Introduction

Dialogue systems (a.k.a. conversational agents) allow people to interact with information through conversations. Dialogue systems play an increasingly important role in society, e.g., as part of personal assistants for online shopping [244] or to support phone calls to service centers [158]. Names of commercial digital assistants – Alexa, Cortana, Google Assistant, Siri – have entered popular culture [76, 155]. And it has been estimated that 80% of businesses had piloted automatic conversational agents by the end of 2021 [151]. The demand for dialogue systems is expected to continue to grow aggressively.

In science, the interest in dialogue systems has also grown rapidly over the past five years. In the artificial intelligence community in general, and the natural language processing and information retrieval communities in particular, dialogue systems and the challenges they bring now play a key role in the research agenda. This is witnessed by (i) a large number of dedicated surveys in natural language processing [2, 20, 79, 129, 151] and information retrieval [56, 57, 82] that are aimed at aggregating findings across large numbers of publications, (ii) a large number of workshops and benchmarks dedicated to dialogue systems such as CAST [17], SCAI [188], and IGLU [80], and (iii) the release of open source toolkits for dialogue systems, such as Convlab-2 [263] and ParlAI [144].

Task-oriented dialogue systems (TDSs) are an important type of dialogue systems. They have raised considerable interest due to their broad applicability, e.g., for booking flight tickets or restaurants, scheduling meetings, and providing medical services [221, 232, 243, 249]. Unlike open-ended dialogue systems [192, 205] for chit-chat, TDSs aim to accurately assist users to achieve specific goals. Besides natural language utterances, TDSs need to deal with predefined goal-related semantic constraints, including intents, slots, states, actions, and frames [45]. An example is shown in Figure 1.1 to illustrate these semantic constraints. An *intent* usually indicates a type of user intention, and a *slot* is an attribute tag for extracting key information as the *value* [228]. A *state* is a distribution over all candidate slot values in an ontology that is used to help interpret the dialogue [70, 179]; it is also known as *belief state* [147] or *user goal* [175]. An *action* usually consists of an action type and several slot-value pairs; it tells the dialogue system how to respond [254]. A *frame* is a combination of the aforementioned semantic constraints [45, 231]; they are the fundamental building blocks for running a TDS.

TDS methods can be divided into two broad categories: *modularized pipeline* TDSs [12, 20, 249] and *end-to-end single-module* TDSs [46, 114, 229], as illustrated

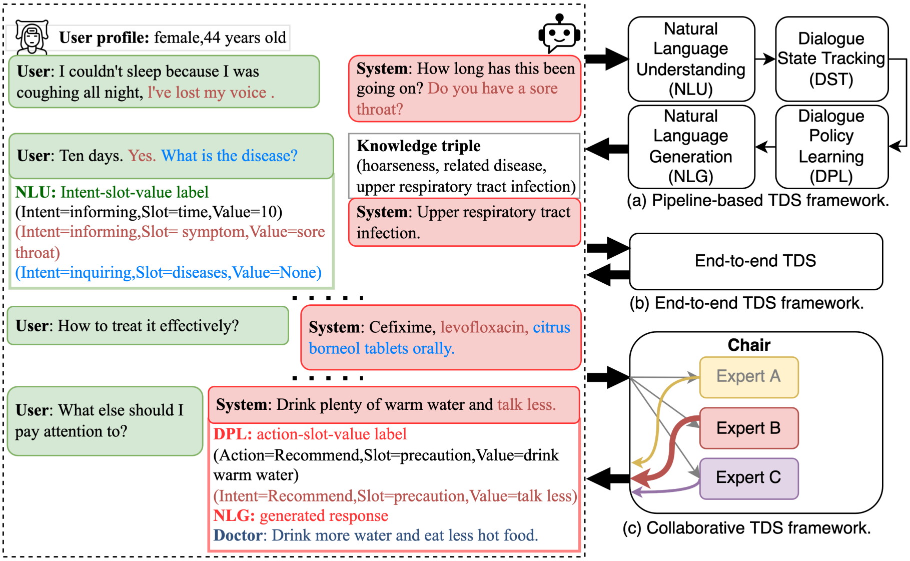

User profile: female,44 years old
User: I couldn't sleep because I was coughing all night, I've lost my voice .
System: How long has this been going on? Do you have a sore throat?
User: Ten days. Yes. What is the disease?
NLU: Intent-slot-value label
(Intent=informing,Slot=time,Value=10)
(Intent=informing,Slot= symptom,Value=sore throat)
(Intent=inquiring,Slot=diseases,Value=None)
Knowledge triple
(hoarseness, related disease, upper respiratory tract infection)
System: Upper respiratory tract infection.
User: How to treat it effectively?
System: Cefixime, levofloxacin, citrus borneol tablets orally.
User: What else should I pay attention to?
System: Drink plenty of warm water and talk less.
DPL: action-slot-value label
(Action=Recommend,Slot=precaution,Value=drink warm water)
(Intent=Recommend,Slot=precaution,Value=talk less)
NLG: generated response
Doctor: Drink more water and eat less hot food.
Natural Language Understanding (NLU)
Dialogue State Tracking (DST)
Natural Language Generation (NLG)
Dialogue Policy Learning (DPL)
(a) Pipeline-based TDS framework.
End-to-end TDS
(b) End-to-end TDS framework.
Chair
Expert A
Expert B
Expert C
(c) Collaborative TDS framework.

Figure 1.1: An example of a task-oriented dialogue (left) and a comparison of task-oriented dialogue system frameworks (right). On the left, the dialogue consists of multiple turns of utterances from the user and the system with goal-related semantic labels. On the right (top), we indicate how a pipeline-based TDS framework functions with four submodules at each dialogue turn: first, NLU outputs intent and slot values; then, DST outputs dialogue states considering previous turns; next, DPL outputs system actions; last, NLG outputs a system response. Also on the right (center), an end-to-end TDS framework directly produces a response based on a single module model. On the right (bottom), a collaborative TDS framework functions with collaborative TDS agents, which leads to better specialization and generalization capabilities.

in Figure 1.1 (a) and (b), respectively. The former decomposes the task-oriented dialogue task into a sequence of subtasks that are addressed by dedicated models for each subtask: (i) natural language understanding (NLU) aims to predict intents and slot-values given a dialogue context (e.g., historical utterances, knowledge base entities) [228]; (ii) dialogue state tracking (DST) aims to predict the updated states in the format of slot-value pairs given a dialogue context and/or the output of NLU [19]; (iii) dialogue policy learning (DPL) aims to predict actions given a dialogue context and/or the output of DST and NLU [246]; and (iv) natural language generation (NLG) aims to generate a response given a dialogue context and/or the output of DPL, DST and NLU [47]. Each module is dependent on one or more upstream modules, which naturally leads to the potential of error propagation between modules and enlarges the effect of the original error [115].

End-to-end single-module TDSs have many attractive characteristics, e.g., global optimization and easier adaptation to new domains [20]. However, it is not always practical to use a single general agent to handle all complex cases in TDSs. For example, an agent that is specialized in booking a restaurant is unlikely to work well in scheduling meetings. Actually, more and more empirical studies from different machine learning applications suggest that no model consistently outperforms all others in all cases [40, 136]. Furthermore, we conducted an early-stage data-driven study to predict a dialogue response in both a selective [167] and generative [166] manner. We found that considering (selective) specification (e.g., information sources, intents) may lead to remarkable differences in the quality of dialogues.

Inspired by this intuition and these pilot results, we pursue to study a new type of *collaborative task-oriented dialogue system* (CTDS) framework, where multiple parallel and/or hierarchical dialogue system agents work in a collaborative manner to achieve better performance than a single, general dialogue system agent. Our *main assumption* is that a team of collaborative, specialized dialogue system agents works better in TDSs than a single general agent, assuming that we are able to design effective learning policies for the agents.

In this thesis we focus on the proposed CTDS framework, which consists of one *chair agent* and several *expert agents*, as shown in Figure 1.1 (c). Each expert agent is specialized for a particular situation, e.g., one domain, one type of action of a system, etc. The chair agent coordinates multiple expert agents in parallel and then adaptively integrates one or several expert agents for the final decision. Compared with existing, end-to-end single-module TDSs, the advantages of CTDSs are three-fold: (i) the specialization of different expert agents and the use of a dynamic chair agent for combining the outputs breaks the bottleneck of a single model; (ii) it is more easily traceable: we can analyze who is to blame when the model makes a mistake; and (iii) by definition, it has a highly parallel character, and it can therefore be implemented in an efficient way.

Under the CTDS framework, we identify three important research directions: (i) how to organize and group dialogue agents in terms of various aspects, e.g., information sources [167], models [166, 168], users [169], and languages [171]; (ii) how to connect different types of dialogue agents, e.g., in a sequential way [170], and or using a chair-expert setup [166–169, 171]; and (iii) how to integrate multiple dialogue agents with appropriate collaboration mechanisms, e.g., retrospective mixture-of-generators and prospective mixture-of-generators [166, 168], hierarchical stochastic attention [170], or incrementally collaborative filtering [169].

In this thesis, we report on research on *collaborative agents for task-oriented dialogue systems*, focusing on the design of the proposed CTDS *framework*, the implementation of specific *models* under the framework, and their applicable *tasks*.

## 1.1 Research outline and questions

As shown in Figure 1.2, we carry out research into collaborative dialogue agents from four angles: (i) *model collaboration* (Chapter 2), where we propose a CTDS framework, and focus on integrating mixture of expert models with a chair model for the NLG task; (ii) *user collaboration* (Chapter 3), where we introduce incremental collaborative filtering over the profiles and dialogues of users to select appropriate dialogue responses; (iii) *language collaboration* (Chapter 4), where we conduct an analytical study of multilingual TDSs based on the mixture-of-languages routing (MOLR) model for the

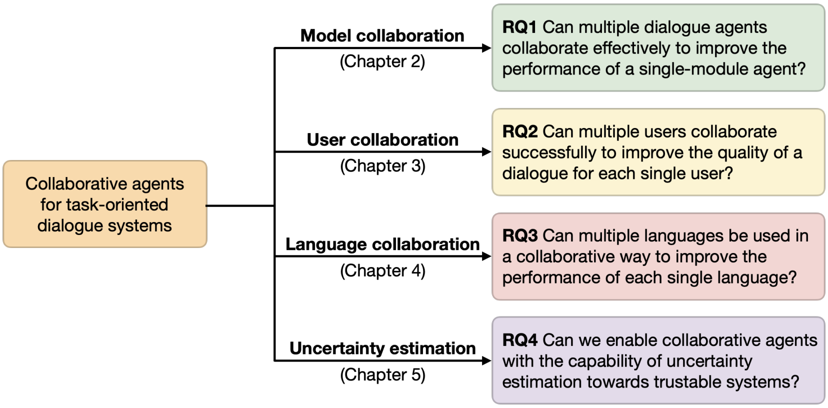

Collaborative agents for task-oriented dialogue systems
Model collaboration
(Chapter 2)
RQ1 Can multiple dialogue agents collaborate effectively to improve the performance of a single-module agent?
User collaboration
(Chapter 3)
RQ2 Can multiple users collaborate successfully to improve the quality of a dialogue for each single user?
Language collaboration
(Chapter 4)
RQ3 Can multiple languages be used in a collaborative way to improve the performance of each single language?
Uncertainty estimation
(Chapter 5)
RQ4 Can we enable collaborative agents with the capability of uncertainty estimation towards trustable systems?

Figure 1.2: A map of the thesis: the research chapters and questions.

NLU and DST tasks; and (iv) *uncertainty estimation* (Chapter 5), where we rethink transformers as sequential collaborative agents. With the aim of building more reliable collaborative agents, we study uncertainty estimation of transformers on three text classification tasks, with both in-domain and out-of-domain settings.

Below, we list our main research questions, each of which corresponds to a particular chapter.

**RQ1** Can multiple dialogue agents collaborate effectively to improve the performance of a single-module agent?

Dialogue response generation (DRG) is a core component of TDSs [47]. Its purpose is to generate proper natural language responses given some context, e.g., historical utterances, system states, etc. State-of-the-art work focuses on how to better tackle DRG in an end-to-end way. Typically, such studies assume that each token is drawn from a single distribution over the output vocabulary, which is not always optimal. For example, on the MultiWOZ dataset [14], we found that the density of the relative token frequency distribution for different intents (i.e., domains, system actions) varies greatly.

To answer **RQ1**, we propose a novel mixture-of-generators network (MoGNet) for DRG, where we assume that each token of a response is drawn from a mixture of distributions. MoGNet consists of a chair generator and several expert generators. Each expert is specialized for DRG w.r.t. a particular intent. The chair collaborates with multiple experts and combines the output they have generated to produce more appropriate responses. We propose two strategies to help the chair make better decisions, namely, a retrospective mixture-of-generators (RMoG) and a prospective mixture-of-generators (PMoG). The former only considers the historical expert-generated responses until the current time step, while the latter also considers possible expert-generated responses in the future by encouraging exploration. In order to differentiate the responsibilities of different experts, we also devise a global-and-local (GL) learning scheme that forces each expert to be specialized towards a particular intent using a local loss and trains the chair and all experts to collaborate using a global loss. We carry out extensive experiments on the MultiWOZ benchmark dataset. MoGNet significantly outperforms state-of-the-art methods in terms of both automatic and human evaluations, demonstrating its effectiveness for DRG.

**RQ2** Can multiple users collaborate successfully to improve the quality of a dialogue for each single user?

There is increasing interest in developing personalized TDSs. Previous work on personalized TDSs often assumes that complete user profiles are available for most or even all users [85, 126, 251]. This is unrealistic because (i) not everyone is willing to expose their profiles due to privacy concerns; and (ii) rich user profiles may involve a large number of attributes (e.g., gender, age, tastes, ...). We assume that similar users collaborate to enrich the context of a user without a complete user profile for personalized TDSs.

To answer **RQ2**, we propose a cooperative memory network (CoMemNN) that has a novel mechanism to gradually enrich user profiles as dialogues progress and to simultaneously improve response selection based on the enriched profiles. CoMemNN consists of two core modules: user profile enrichment (UPE) and dialogue response selection (DRS). The former enriches incomplete user profiles by utilizing collaborative information from neighbor users as well as current dialogues. The latter uses the enriched profiles to update the current user query so as to encode more useful information, based on which a personalized response to a user request is selected. We conduct extensive experiments on the personalized bAbI dialogue benchmark datasets. We found that CoMemNN is able to enrich user profiles effectively, with robustness, which results in a continuous improvement of response selection accuracy compared to state-of-the-art methods.

**RQ3** Can multiple languages be used in a collaborative way to improve the performance of each single language?

More than 6,900 languages are widely used for global services and communication in international markets and communities [53]. The demand to cross the chasms of multilingual conversations calls for research on multilingual TDSs [181]. However, multilingual TDS models face three main challenges: (i) A data acquisition dilemma. It is expensive and tedious to collect high-quality task-oriented (monolingual) conversations with fine-gained labels [243], and this is even worse for multilingual conversations. The acquisition of non-English data is more challenging due to a lack of expertise in low-resource languages. For example, mBERT is trained on Wikipedia articles, where English has the largest volume of data (15.5 GB) while the low-resource language Yoruba has the smallest data volume (10 MB) [237]. (ii) Global optimization of multilingual models. Most recent work into multilingual conversations optimize either monolingual models or they perform crosslingual. Only a few studies consider simultaneous multilingual performance [146, 148]. And (iii) The influence of language commonalities and peculiarities. It is non-trivial to learn multilingual models due to the diverse characteristics of multiple languages [3].

To answer **RQ3**, we conduct an analytical study regarding language characteristics in collaborative TDSs, which aims to facilitate the research into multilingual TDSs from a new angle. First, we implement a simple yet effective mixture-of-languages routing

(MOLR) model as a benchmark, which transforms the multilingual TDS problem into a collaborative TDS problem. An expert agent can be either a monolingual dialogue agent or a cross-lingual dialogue agent, and the chair agent conducts global optimization among multiple so-called language routes through collaboration policies. We implement these ideas based on the state-of-the-art pre-trained language model mT5 [242] for each expert agent, and introduce two collaboration policies (i.e., parameter sharing and language-route addressing). Then, we analyze the influential factors in multilingual TDSs in terms of both the language and model aspects. We hope that our analytical study inspires the design of multilingual TDS models and helps to speed up progress on the problem of multilingual TDS.

**RQ4** Can we enable collaborative agents with the capability of uncertainty estimation towards trustworthy systems?

Transformers and their variants have been the state-of-the-art basis for many NLP and dialogue tasks. The vanilla transformer [216] consists of a stack of transformer blocks, each of which can be seen as a single agent. Intuitively, we can think of transformers as a sequence of collaborative agents. Understanding the reliability and certainty of transformer model predictions is crucial for building trustworthy machine learning applications, e.g., for medical diagnosis. Although many recent transformer extensions have been proposed, the study of the uncertainty estimation of transformer models is under-explored.

To answer **RQ4**, we propose a novel way to enable transformers to do uncertainty estimation and, meanwhile, retain the original predictive performance. This is achieved by learning a hierarchical stochastic self-attention mechanism that attends to values and a set of learnable centroids, respectively. Then, new attention heads are formed with a mixture of sampled centroids using the Gumbel-Softmax trick. We theoretically show that the self-attention approximation by sampling from a Gumbel distribution is upper bounded. We empirically evaluate our model on three text classification tasks with both in-domain (ID) and out-of-domain (OOD) datasets. The experimental results demonstrate that our approach (i) achieves the best predictive performance and uncertainty trade-off among compared methods; (ii) exhibits very competitive (in most cases, improved) predictive performance on ID datasets; and (iii) is on par with Monte Carlo dropout and ensemble methods in uncertainty estimation on OOD datasets.

## 1.2 Main contributions

In this section, we list the main contributions of this thesis: theoretical contributions, algorithmic contributions, empirical contributions, and resource contributions.

### 1.2.1 Theoretical contributions

- A novel chair-experts task-oriented dialogue system framework with a chair agent and multiple expert agents, where multiple parallel and/or hierarchical agents collaboratively work together to achieve better performance than a single agent (Chapter 2).

- A theoretical proof that the error of stochastic attention approximation in the hierarchical stochastic transformer has an upper bound (Chapter 5).

### 1.2.2 Algorithmic contributions

- A mixture-of-generators network (MoGNet) model, where the chair makes good decisions based on retrospective and prospective strategies (i.e., RMoG and PMoG) and a learning scheme (i.e., GL) (Chapter 2).

- A cooperative memory network (CoMemNN) model that can gradually enrich user profiles with collaborative users as dialogues progress, and simultaneously improve the performance of response selection based on the enriched profiles (Chapter 3).

- A learning algorithm for multiple hop CoMemNN (Chapter 3).

- A statistic $\sigma$, namely stability coefficient, which is defined as the standard deviation of a list of performance results, and can be used to evaluate model stability (Chapter 3).

- A mixture-of-languages routing (MOLR) model, where each expert agent can be either a monolingual or cross-lingual dialogue agent, and the chair agent conducts global optimization among multiple language routes by collaboration policies (i.e., parameter sharing and language-path routing) (Chapter 4).

- Hierarchical stochastic transformer models (i.e., STO-TRANS, H-STO-TRANS), where stochasticity is introduced into the self-attention mechanism in transformers to provide uncertainty information with predictions (Chapter 5).

### 1.2.3 Empirical contributions

- A statistical study of token distribution on the Multi-domain Wizard-of-Oz (MultiWOZ) dataset, which motivates the hypothesis that a response should be drawn from a mixture of distributions for multiple intents rather than from a single distribution for a general intent (Chapter 2).

- An empirical verification of the effectiveness of model collaboration using the MoGNet model for DRG task (Chapter 2).

- An empirical verification of the effectiveness of user collaboration using the CoMemNN model for personalized TDS with incomplete user profiles (Chapter 3).

- An empirical verification of the effectiveness of language collaboration using the MOLR model for multilingual TDS task (Chapter 4).

- An empirical verification of the effectiveness of uncertainty estimation in stochastic transformer models for both ID and OOD tasks (Chapter 5).

### 1.2.4 Resource contributions

- The source code of MoGNet model is released under an open source license (Chapter 2).
- The source code of CoMemNN model is released under an open source license (Chapter 3).
- The source code of stochastic transformer models is released under an open source license (Chapter 5).
- A simulated dataset for personalized TDS with incomplete user profiles, which is generated by randomly dropout user profiles with different ratios (Chapter 3).

## 1.3 Thesis overview

In this thesis, we focus on collaborative agents in task-oriented dialogue systems. Below, we provide an overview of each chapter.

In this chapter, we identify the demands of dialogue systems and provide some background knowledge, particularly concerning the tasks and models in TDSs. We also discuss the motivation of the CTDS framework and an implementation of the framework with a chair-experts architecture. Finally, we summarize what we bring to the three main research directions in the under-explored area of collaborative agents for task-oriented dialogue systems.

Next, in Chapter 2, we model collaboration for collaborative task-oriented dialogue system (CTDS). First, we propose a chair-experts framework under the CTDS framework and introduce a mixture-of-generators network (MoGNet) model with collaborative agents for the dialogue response generation (DRG) task. Then, we empirically compare MoGNet with state-of-the-art single agents, and examine their performance in terms of both automatic and human evaluation. We also explore the impact of different collaboration policies and learning schemes on MoGNet. Finally, we conduct an analytical study on partition intents, influential hyper-parameters, and example cases with predictions from MoGNet and the models used for comparison. The experimental results demonstrate that MoGNet outperforms single-agent models against which we compare thanks to the RMoG and PMoG mechanisms, as well as the GL learning scheme.

In Chapter 3, we consider user collaboration for CTDS. First, we propose a practical task “personalized TDSs with incomplete user profiles.” Then, we propose a CoMemNN model that consists of two cooperative modules: (i) a UPE module that gradually learns to enrich user profiles from collaborative users, and (ii) a DRS module that improves the performance of response selection based on enriched user profiles. Next, we empirically compare CoMemNN and the state-of-the-art baselines with and without discarding user profiles. After that, we conduct an ablation study for the UPE and DRS modules, and further analyze the effect of a multiple-hop mechanism and profile attributes. The experimental results demonstrate that CoMemNN can effectively enrich user profiles as well as select high-quality responses based on a cooperative mechanism and a multiple-hop learning algorithm.

In Chapter 4, we study language collaboration for CTDS. First, we introduce how to model full TDS in a unified fashion and implement state-of-the-art benchmarks, respectively. Specifically, we develop: (i) multiple expert agents based on the mT5 model for monolingual and cross-lingual TDSs, respectively; (ii) a multilingual TDS based on fine-tuning the mT5 model on multilingual data with a parameter sharing mechanism; and (iii) a multilingual TDS under the chair-experts framework, namely MOLR, where a chair agent learns a mixture of expert agents for multiple language routes by different collaboration policies (i.e., parameter sharing and language-route addressing). Then, we dive into an analytical study in terms of language and model aspects, which, we hope, informs future work in the multilingual TDS research community. The experimental results demonstrate that language-specific characteristics have a great impact on multilingual TDSs, and collaborative agents (e.g., MOLR) have the potential to improve multilingual TDSs based on monolingual and cross-lingual agents.

Finally, in Chapter 5, we study uncertainty estimation for collaborative agents. First, we introduce background material on predictive uncertainty and the vanilla transformer model. Then, we describe how to model uncertainty with Bayesian inference theory for classification tasks. Next, we propose stochastic transformer models (i.e., STO-TRANS, H-STO-TRANS) by introducing Gumbel-Softmax to hierarchical attention in the vanilla transformer. After that, we conduct empirical ID and OOD studies on three classification tasks, i.e., sentiment analysis (SA), linguistic acceptability (LA) and slot filling (SF). The experimental results demonstrate that hierarchical stochastic transformers can provide effective predictions with uncertainty estimation, and learn good trade-offs of performance between in-domain prediction and out-of-domain uncertainty estimation.

Finally, in Chapter 6, we conclude the thesis and discuss potential directions for future work.

## 1.4 Origins

In this section, we list the publications that form the basis for each chapter and briefly explain the role of each author.

**Chapter 2** is based on the following paper:

- J. Pei, P. Ren, and M. de Rijke. A modular task-oriented dialogue system using a neural mixture-of-experts. In *SIGIR Workshop on Conversational Interaction Systems*, 2019.
- J. Pei, P. Ren, C. Monz, and M. de Rijke. Retrospective and prospective mixture-of-generators for task-oriented dialogue response generation. In *ECAI*, pages 2148–2155, 2020.

JP designed the model, implemented it, ran the experiments, and analyzed most results. PR and MdR helped with the model design and technical details. PR and CM helped with intermediate result analysis and the design of the final experimental setup. All authors contributed to the writing.

**Chapter 3** is based on the following paper:

- J. Pei, P. Ren, and M. de Rijke. A cooperative memory network for personalized task-oriented dialogue systems with incomplete user profiles. In *The Web Conference*, pages 1552–1561, 2021.

JP designed the model, implemented it, ran the experiments, and analyzed most results. PR helped with the model design and technical details. PR and MdR helped with intermediate result analysis and the design of the final experimental setup. All authors contributed to the writing.

**Chapter 4** is based on the following paper:

- J. Pei, G. Yan, P. Ren, and M. de Rijke. Mixture-of-languages routing for multilingual task-oriented dialogue systems. *Under review*, 2022.

JP designed the model, implemented it, ran the experiments, and analyzed most results. GY helped with implementing the model and running experiments. PR helped with the model design and technical details. PR and MdR helped with intermediate result analysis and the design of the final experimental setup. All authors contributed to the writing.

**Chapter 5** is based on the following paper:

- J. Pei, C. Wang, and G. Szarvas. Transformer uncertainty estimation with hierarchical stochastic attention. In *AAAI*, pages 11147–11155, 2022.

JP designed the model, implemented it, ran most experiments, and analyzed most results. CW helped with the model design and technical details. CW and GS helped with intermediate result analysis and the design of the final experimental setup. All authors contributed to the writing.

The writing of the thesis also benefited from work on the following publications:

- J. Pei, A. Stienstra, J. Kiseleva, and M. de Rijke. SEntNet: Source-aware recurrent entity network for dialogue response selection. In *IJCAI Workshop SCAI*, 2019.
- G. Yan, J. Pei, P. Ren, Z. Ren, X. Xin, H. Liang, M. de Rijke, and Z. Chen. ReMeDi: Resources for multi-domain, multi-service, medical dialogues. In *SIGIR*, pages 3013–3024, 2022.
- B. Wang, Q. Xie, J. Pei, P. Tiwari, Z. Li, and J. Fu. Pre-trained language models in biomedical domain: a systematic survey. *Association for Computing Machinery*, 2021.

# 2 Model Collaboration: Mixture-of-Generators for Response Generation

In this chapter, we aim to answer the following research question:

**RQ1** Can multiple dialogue agents collaborate effectively to improve the performance of a single-module agent?

We explore the idea of collaborative agents from the point of view of model collaboration. We first propose a collaborative chair-experts framework, and a mixture-of-generators network for the dialogue response generation (DRG) task. Our main finding about this framework is that MoGNet outperforms single-module models, which is dependent on the appropriate partition, topological construction of dialogue agents as well as effective collaboration policies.

## 2.1 Introduction

Task-oriented dialogue systems (TDSs) have sparked considerable interest due to their broad applicability, e.g., for booking flight tickets or scheduling meetings [232, 249]. In TDSs, there are many factors to consider in order to achieve good performance, such as user intent understanding [229], dialogue state tracking [260], and dialogue response generation (DRG) [14]. Given a *dialogue context* (dialogue history, states, retrieved results from a knowledge base, etc.), the purpose of DRG is to generate a proper natural language response that leads to task completion, i.e., successfully achieving specific goals, and that is fluent, i.e., generating natural and fluent utterances.

Recently proposed DRG methods have achieved promising results. For example, S2SAttnLSTM [13, 14] follows the dominant Sequence-to-Sequence (Seq2Seq) model under an encoder-decoder architecture; LaRLAttnGRU [256] treats action spaces as latent variables for reinforcement learning. However, when generating a response,

This chapter was published as J. Pei, P. Ren, and M. de Rijke. A modular task-oriented dialogue system using a neural mixture-of-experts. In *SIGIR Workshop on Conversational Interaction Systems*, 2019 [166] and J. Pei, P. Ren, C. Monz, and M. de Rijke. Retrospective and prospective mixture-of-generators for task-oriented dialogue response generation. In *ECAI*, pages 2148–2155, 2020 [168].

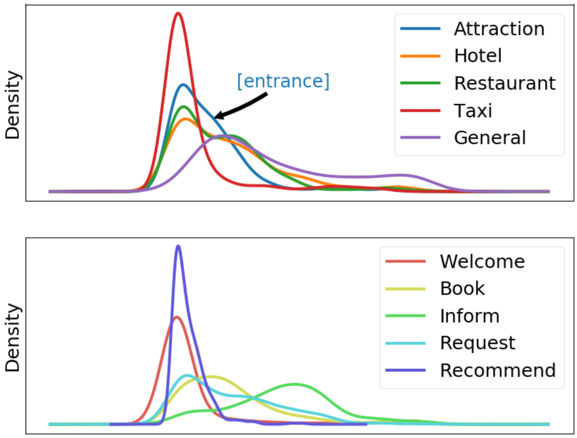


Figure 2.1: Density of the relative token frequency distribution for different intents (*domains* in the top plot, *system actions* in the bottom plot). We use kernel density estimation[1] to estimate the probability density function of a random variable from a relative token frequency distribution.

all current models assume that each token is drawn from a single distribution over the output vocabulary. This may be unreasonable because responses vary greatly with different intents, where intents may refer to domains, system actions, or other criteria for partitioning responses, e.g., the source of dialogue context [167]. To support this claim, consider the training set of the Multi-domain Wizard-of-Oz (MultiWOZ) benchmark dataset [14], where 67.4% of the dialogues span across multiple domains, and all the dialogues span across multiple types of system actions. We plot the density of the relative token frequency distributions in responses of different intents over the output vocabulary in Figure 2.1. Although there is some overlap among distributions, there are also clear differences. For example, when generating the token $[entrance]$, it has a high probability of being drawn from the distributions for the intent of *booking an attraction*, but not from *booking a taxi*. Thus, we hypothesize that a response should be drawn from a mixture of distributions for multiple intents rather than from a single distribution for a general intent.

We adopt a model collaboration point of view and propose a *mixture-of-generators network* (MoGNet) for DRG, which consists of a *chair* generator and several *expert* generators. Each expert model is specialized to generate a token from a single distribution for a particular intent, e.g., one domain, one type of action of a system, etc. The chair model collaborates with multiple expert models and generates the final response by taking the utterances generated by the expert models into consideration. Compared with previous methods, the advantages of MoGNet are at least two-fold: First, the specialization of different experts and the use of a chair for combining the outputs breaks the bottleneck of a single model [40, 136]. Second, it is more easily *traceable*:

[1] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.kde.html

we can analyze who is responsible when the model makes a mistake and generates an inappropriate response.

We propose two strategies to help the chair make good decisions, i.e., *retrospective mixture-of-generators* (RMoG) and *prospective mixture-of-generators* (PMoG). RMoG only considers the retrospective utterances generated by the experts, i.e., the utterances generated by all the experts prior to the current time step. However, a chair without a long-range vision is likely to make sub-optimal decisions. Consider, for example, these two responses: “what **day** will you be traveling?” and “what **day** and **time** would you like to travel?” If we only consider these responses until the 2nd token (which RMoG does), then the chair might choose the first response due to the absence of a more long-range view of the important token “time” located after the 2nd token. Hence, we also propose a PMoG, which enables the chair to make full use of the prospective predictions of experts as well.

To effectively train MoGNet, we devise a *global-and-local* (GL) learning scheme. The local loss is defined on a segment of data with a certain intent, which forces each expert to specialize. The global loss is defined on all data, which forces the chair and all experts to collaborate with each other. The global loss can also improve data utilization by enabling the backpropagation error of each data sample to influence all experts as well as the chair.

To verify the effectiveness of MoGNet, we carry out experiments on the MultiWOZ benchmark dataset. MoGNet significantly outperforms state-of-the-art DRG methods, improving over the best performing model on this dataset by 5.64% in terms of overall performance (0.5**Inform*+0.5**Success*+*BLEU*) and 0.97% in terms of response generation quality (*Perplexity*).

The main contributions of this chapter are:

- A novel MoGNet model that is the first framework that devises chair and expert generators for DRG, to the best of our knowledge;
- Two novel collaboration policies, i.e., RMoG and PMoG, to help the chair make better decisions; and
- A GL learning scheme to differentiate experts and fuse data efficiently.

## 2.2 Related work

In this section, we summarize recent work under two dominant frameworks for TDSs: modularized pipeline TDSs and end-to-end single-module TDSs. Then we compare related work on DRG.

### 2.2.1 Modularized pipeline TDSs

Modularized pipeline TDS frameworks consist of a pipeline with several modules. Examples include natural language understanding (NLU) [7, 21], dialogue state tracking (DST) [180, 260], dialogue policy learning (DPL) [254], and natural language generation (NLG) [44, 143, 247]. Each module has an explicitly decomposed function for a specialized subtask, which is beneficial for tracking errors. Young et al. [249] summarize typical pipeline TDSs that are constitutive of distinct modules following a POMDP paradigm. Crook et al. [30] develop a TDS platform that is loosely decomposed into three modules, i.e., initial processing of input, dialogue state updates, and policy execution. Yan et al. [244] present a TDS for completing various purchase-related tasks by optimizing individual upstream-dependent modules, i.e., query understanding, state tracking, and dialogue management. However, the pipeline setting of these methods will unavoidably run into the upstream propagation problem [20], the module interdependence problem [20], and the joint evaluation problem [249].

Unlike the methods listed above, our MoGNet consists of a group of modules, including a chair bot and several expert bots. This design addresses the module interdependence problem since each module is independent of the others. Besides, the chair bot alleviates the error propagation problem because it is able to manage the overall errors through an effective learning scheme.

### 2.2.2 End-to-end single-module TDSs

End-to-end single-module systems address the TDS task with only one module, which directly maps a *dialogue context* to a *response* [229]. There is a growing focus in research on end-to-end approaches for TDSs, which can enjoy global optimization and facilitate easier adaptation to new domains [20]. Sordoni et al. [202] show that using a recurrent neural network (RNN) to generate text conditioned on the dialogue history results in more natural conversations. Later improvements have been made by adding an attention mechanism [106, 217], by modeling the hierarchical structure of dialogues [192], or by jointly learning belief spans [103]. However, existing studies on end-to-end TDSs mostly use a single-module model to generate responses for complex dialogue contexts. This is practically problematic because dialogue contexts are very complicated with multiple sources of information [21]. In addition, previous studies show that it is rare to find a single model that achieves the best results on the overall task based on empirical studies from different machine learning applications [40, 136].

Different from the methods listed above, which use a single module to address TDSs, our MoGNet uses multiple modules (expert and chair bots), which makes good use of the specialization of different experts and the generalization of the chair for combining the final outputs. Besides, our MoGNet model is able to track who is to blame when the model makes a mistake.

### 2.2.3 Dialogue response generation

Recent work views DRG as a source-to-target transduction problem, which maps a *dialogue context* to a *response* [46, 107, 229]. Sordoni et al. [202] show that using an RNN to generate text conditioned on dialogue history results in more natural conversations. Later improvements include the addition of attention mechanisms [106, 217], modeling the hierarchical structure of dialogues [192], or jointly learning belief spans [103]. Strengths of these methods include global optimization and easier adaptation to new domains [20].

The studies listed above assume that each token of a response is sampled from a single distribution, given a complex dialogue context. In contrast, MoGNet uses multiple cooperating modules, which exploits the specialization capabilities of different experts and the generalization capability of a chair. Work most closely related to ours in terms of modeling multiple experts includes [24, 65, 100, 166]. Le et al. [100] integrate a chat model with a question-answering model using an LSTM-based mixture-of-experts method. Their model is similar to MoGNet-GL-P (without PMoG and GL) except that they simply use two implicit expert generators that are not specialized on particular intents. Guo et al. [65] introduce a mixture-of-experts to use the data relationship between multiple domains for binary classification and sequence tagging. Sequence tagging generates a set of fixed labels; DRG generates diverse appropriate response sequences.

The differences between MoGNet and these two approaches are three-fold: First, MoGNet consists of a group of modules, including a chair generator and several expert generators; this design addresses the module interdependence problem since each module is independent of the others. Second, the chair generator alleviates the error propagation problem because it is able to manage the overall errors through an effective learning scheme. Third, the models of those two approaches cannot be directly applied to task-oriented DRG. The recently published HDSA [24] slightly outperforms MoGNet on *Score* (+0.07), but it overly relies on BERT [39] and graph structured dialog acts. MoGNet follows the same modular TDS framework [166], but it performs substantially better due to fitting the expert generators with both retrospection and prospection abilities and adopting the GL learning scheme to conduct more effective learning.

## 2.3 Mixture-of-generators network

We focus on task-oriented DRG (a.k.a. the context-to-text generation task [14]). Formally, given a current dialogue context $X = (U, B, D)$, where $U$ is a combination of previous utterances, $B$ are the belief states, and $D$ are the retrieved database results based on $B$, the goal of task-oriented DRG is to generate a fluent natural language response $Y = (y_1, \dots, y_n)$ that contains appropriate system actions to help users accomplish their task goals, e.g., booking a flight ticket. We propose MoGNet to model the generation probability $P(Y \mid X)$.

### 2.3.1 Overview

The MoGNet framework consists of two types of roles:

- $k$ **expert generators**, each of which is specialized for a particular *intent*, e.g., a domain, a type of action of a system, etc. Let $\mathcal{D} = \{(X_p, Y_p)\}_{p=1}^{|\mathcal{D}|}$ denote a dataset with $|\mathcal{D}|$ independent samples of $(X, Y)$. Expert-related intents partition $\mathcal{D}$ into $k$ pieces $\mathcal{S} = \{\mathcal{S}_l\}_{l=1}^{k}$, where $\mathcal{S}_l \triangleq \{(X_p^l, Y_p^l)\}_{p=1}^{|\mathcal{S}_l|}$. Then $\mathcal{S}_l$ is used to train each expert by predicting $P^l(Y^l \mid X^l)$. We expect the $l$-th expert to perform better than the others on $\mathcal{S}_l$.
- a **chair generator**, which learns to collaborate with a group of experts to make an optimal decision. The chair is trained to predict $P(Y \mid X)$, where $(X, Y)$ is a

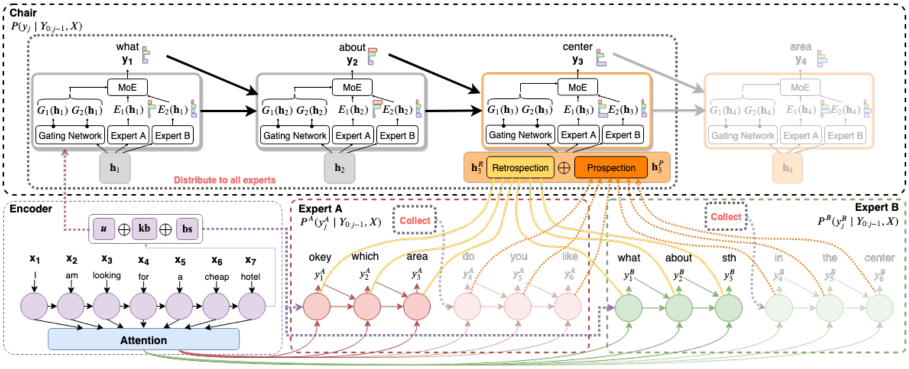

Chair
$P(y_j \mid Y_{0:j-1}, X)$
what
$\mathbf{y_1}$
about
$\mathbf{y_2}$
center
$\mathbf{y_3}$
area
$\mathbf{y_4}$
MoE
$G_1(\mathbf{h}_1)$ $G_2(\mathbf{h}_1)$ $E_1(\mathbf{h}_1)$ $E_2(\mathbf{h}_1)$
$G_1(\mathbf{h}_2)$ $G_2(\mathbf{h}_2)$ $E_1(\mathbf{h}_2)$ $E_2(\mathbf{h}_2)$
$G_1(\mathbf{h}_3)$ $G_2(\mathbf{h}_3)$ $E_1(\mathbf{h}_3)$ $E_2(\mathbf{h}_3)$
$G_1(\mathbf{h}_4)$ $G_2(\mathbf{h}_4)$ $E_1(\mathbf{h}_4)$ $E_2(\mathbf{h}_4)$
Gating Network
Expert A
Expert B
$\mathbf{h}_1$
$\mathbf{h}_2$
$\mathbf{h}_4$
$\mathbf{h}_3^R$
Retrospection
Prospection
$\mathbf{h}_3^P$
Distribute to all experts
Encoder
$u$ kb bs
$\mathbf{x_1}$ $\mathbf{x_2}$ $\mathbf{x_3}$ $\mathbf{x_4}$ $\mathbf{x_5}$ $\mathbf{x_6}$ $\mathbf{x_7}$
I am looking for a cheap hotel
Attention
Expert A
$P^A(y_j^A \mid Y_{0:j-1}, X)$
Collect
okey which area do you like
$y_1^A$ $y_2^A$ $y_3^A$ $y_4^A$ $y_5^A$ $y_6^A$
Expert B
$P^B(y_j^B \mid Y_{0:j-1}, X)$
Collect
what about sth in the center
$y_1^B$ $y_2^B$ $y_3^B$ $y_4^B$ $y_5^B$ $y_6^B$

Figure 2.2: Overview of MoGNet. It illustrates how the model generates the token $y_3$ given sequence $X$ as an input in the process of generating the whole sequence $Y$ as a dialogue response.

sample from $\mathcal{D}$.

Figure 2.2 shows our implementation of MoGNet; it consists of three types of components, i.e., a shared context encoder, $k$ expert decoders, and a chair decoder. For example, for the third token $y_3$, the chair can collaboratively consider $[h_3^R, h_3^P]$, the hidden states of generated utterances, from both expert A ("okay which area do you like", i.e., $[y_1^A, \ldots, y_6^A]$) and expert B ("what about something in the center", i.e., $[y_1^B, \ldots, y_6^B]$). The chair can foresee the prediction after the third timestamps of experts, and the prospective token "center" from expert B helps with the final prediction. Next, we expand on each component, respectively.

### 2.3.2 Shared context encoder

The role of the shared context encoder is to read the dialogue context $X$ and construct a representation. We follow Budzianowski et al. [13] and model the current dialogue context as a combination of user utterances $U$, belief states $B$, and retrieval results from a database $D$.

First, we employ a RNN [25] to map a sequence of input tokens $U = \{w_1, \ldots, w_m\}$ to hidden vectors $\mathbf{H}^U = \{\mathbf{h}_1^U, \ldots, \mathbf{h}_m^U\}$. The hidden vector $\mathbf{h}_i$ at the $i$-th step can be represented as:

$$\mathbf{h}_i^U, \mathbf{s}_i = \mathrm{RNN}(\mathbf{w}_i, \mathbf{h}_{i-1}^U, \mathbf{s}_{i-1}), \tag{2.1}$$

where $\mathbf{w}_i$ is the embedding of the token $w_i$. The initial state $\mathbf{s}_0$ of the RNN is set to 0.

Then, we represent the current dialogue context $\mathbf{x}$ as a combination of the user utterance representation $\mathbf{h}_m^U$, the belief state vector $\mathbf{h}^B$, and the database vector $\mathbf{h}^D$:

$$\mathbf{x} = \tanh(\mathbf{W}_u \mathbf{h}_m^U + \mathbf{W}_b \mathbf{h}^B + \mathbf{W}_d \mathbf{h}^D), \tag{2.2}$$

where $\mathbf{h}_m^U$ is the final hidden state from Eq. 2.1; $\mathbf{h}^B$ is a 0-1 vector with each dimension representing a state (slot-value pair); $\mathbf{h}^D$ is also a 0-1 vector, which is built by querying the database with the current state $B$. Each dimension of $\mathbf{h}^D$ represents a particular result from the database (e.g., whether a flight ticket is available).

### 2.3.3 Expert decoder

Given the current dialogue context $X$ and the current decoded tokens $Y_{0:j-1}$, the $l$-th expert outputs the probability $P^l(y^l_j \mid Y_{0:j-1}, X)$ over the vocabulary $\mathcal{V}$ at the $j$-th step by:

$$\begin{aligned} P^l(y^l_j \mid Y_{0:j-1}, X) &= \mathrm{softmax}(\mathbf{U}^T \mathbf{o}^l_j + \mathbf{b}) \\ \mathbf{o}^l_j, \mathbf{s}^l_j &= \mathrm{RNN}(\mathbf{y}_{j-1} \oplus \mathbf{c}^l_j, \mathbf{o}^l_{j-1}, \mathbf{s}^l_{j-1}), \end{aligned} \tag{2.3}$$

where $\mathbf{U}$ is the parameter matrix and $\mathbf{b}$ is the bias; $\mathbf{s}^l_j$ is the state vector, which is initialized by the dialogue context vector from the shared context encoder, i.e., $\mathbf{s}^l_0 = \mathbf{x}$; $\mathbf{y}_{j-1}$ is the embedding of the generated token at time step $j-1$; $\oplus$ is the concatenation operation; $\mathbf{c}^l_j$ is the context vector which is calculated with a concatenation attention mechanism [5, 127] over the hidden representations from a shared context encoder as follows:

$$\begin{aligned} \mathbf{c}^l_j &= \sum_{i=1}^{m} \alpha^l_{ji} \mathbf{h}_i \\ \alpha^l_{ji} &= \frac{\exp(w^l_{ji})}{\sum_{i=1}^{m} \exp(w^l_{ji})} \\ w^l_{ji} &= \mathbf{v}^T_l \tanh\left(\mathbf{W}^T_l (\mathbf{h}_i \oplus \mathbf{s}^l_{j-1}) + \mathbf{b}_l\right), \end{aligned} \tag{2.4}$$

where $\alpha$ is a set of attention weights; $\oplus$ is the concatenation operation. $\mathbf{W}_l$, $\mathbf{b}_l$, $\mathbf{v}_l$ are learnable parameters, which are not shared by different experts in our experiments.

### 2.3.4 Chair decoder

Given the current dialogue context $X$ and the current decoded tokens $Y_{0:j-1}$, the chair decoder estimates the final token prediction distribution $P(y_j \mid Y_{0:j-1}, X)$ by combining the prediction probabilities from $k$ experts. Here, we consider two strategies to leverage the prediction probabilities from experts, i.e., RMoG and PMoG. The former only considers expert generator outputs from *history* (until the $(j-1)$-th time step), which follows the typical neural Mixture-of-Experts (MoE) architecture [191, 195]. We propose the latter to make the chair generator envision the*future* (i.e., after the $(j-1)$-th time step) by exploring expert generator outputs from $t$ extra steps ($t \in [1, n-j], t \in \mathbb{N}$).

Specifically, the chair determines the prediction $P(y_j \mid Y_{0:j-1}, X)$ as follows:

$$\begin{aligned} P(y_j \mid Y_{0:j-1}, X) = {} & \beta^C_j \cdot P(y^c_j \mid Y_{0:j-1}, X) \\ & + \sum_{l=1}^{k} (\beta^{l,R}_j + \beta^{l,P}_j) \cdot P(y^l_j \mid Y^l_{:j-1}, X), \end{aligned} \tag{2.5}$$

where $P(y^c_j \mid Y_{0:j-1}, X)$ is the prediction probability from the chair itself; $P(y^l_j \mid Y_{0:j-1}, X)$ is the prediction probability from expert $l$; $\beta_j = [\beta^C_j, \beta^{l,R}_j, \beta^{l,P}_j]$ are normalized collaboration coefficients, which are calculated as:

$$\begin{aligned}\beta_j &= \frac{\exp(\mathbf{v}^T \mathbf{h}_j)}{\sum_{l=1}^{k} \exp(\mathbf{v}^T \mathbf{h}_l)} \\ \mathbf{h}_j &= \text{MLP}([P(y_j^c \mid Y_{0:j-1}, X), \mathbf{h}_j^R, \mathbf{h}_j^P]).\end{aligned} \tag{2.6}$$

$\beta_j^C$, $\beta_j^{l,R}$ and $\beta_j^{l,P}$ are estimated w.r.t. $P(y_j^c \mid Y_{0:j-1}, X)$, $\mathbf{h}_j^R$ and $\mathbf{h}_j^P$, respectively. $\mathbf{h}_j^R$ is a list of retrospective decoding outputs from all experts, which is defined as follows:

$$\begin{aligned}\mathbf{h}_j^R = {} & P(y_{1:j-1}^1 \mid y_0, X) \oplus \cdots \oplus P(y_{1:j-1}^l \mid y_0, X) \\ & \oplus P(y_{1:j-1}^k \mid y_0, X),\end{aligned} \tag{2.7}$$

where $y_0$ is a special token “[BOS]” indicating the start of decoding; $P(y_{1:j-1}^l \mid y_0, X)$ is the output of expert $l$ from the 1-st to the $(j-1)$-th step using Eq. 2.3; $\mathbf{h}_j^P$ is a list of prospective decoding outputs from all experts, which is defined as follows:

$$\begin{aligned}\mathbf{h}_j^P = {} & P(y_{j:j+t}^1 \mid Y_{0:j-1}, X) \oplus \cdots \\ & \oplus P(y_{j:j+t}^l \mid Y_{0:j-1}, X) \\ & \oplus P(y_{j:j+t}^k \mid Y_{0:j-1}, X),\end{aligned} \tag{2.8}$$

where $P(y_{j:j+t}^l \mid Y_{0:j-1}, X)$ are the outputs of expert $l$ from the $j$-th to $(j+t)$-th step. We obtain $P(y_{j:j+t}^l \mid X)$ by forcing expert $l$ to generate $t$ steps using Eq. 2.3 based on the current generated tokens $Y_{0:j-1}$.

### 2.3.5 Learning scheme

We devise a global-and-local learning scheme to train MoGNet. Each expert $l$ is optimized by a localized expert loss defined on $\mathcal{S}_l$, which forces each expert to specialize on one of the portions of data $\mathcal{S}_l$. We use cross-entropy loss for each expert and the joint loss for all experts is as follows:

$$\mathcal{L}_{experts} = \sum_{l=1}^{k} \sum_{(X_p^l, Y_p^l) \in \mathcal{S}_l} \sum_{j=1}^{n} \mu_l y_j^l \log P(y_j^l \mid Y_{0:j-1}^l, X), \tag{2.9}$$

where $P(y_j^l \mid Y_{0:j-1}^l, X)$ is the token prediction by expert $l$ (Eq. 2.3) computed on the $r$-th data sample; $y_j^l$ is a one-hot vector indicating the ground truth token at $j$.

We also design a global chair loss to differentiate the losses incurred by different experts. The chair can attribute the source of errors to the expert in charge. For each data sample in $\mathcal{D}$, we calculate the combined taken prediction $P(y_j \mid Y_{0:j-1}, X)$ (Eq. 2.5). Then the global loss becomes:

$$\mathcal{L}_{chair} = \sum_{r=1}^{|\mathcal{D}|} \sum_{j=1}^{n} y_j \log P(y_j \mid Y_{0:j-1}, X). \tag{2.10}$$

Our overall optimization follows the joint learning paradigm that is defined as a weighted combination of constituent losses:

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{experts} + (1 - \lambda) \cdot \mathcal{L}_{chair}, \tag{2.11}$$

where $\lambda$ is a hyper-parameter to regulate the importance between the experts and the chair for optimizing the loss.

## 2.4 Experimental setup

In this subsection, we summarize our research questions, the dataset used for our experiments, the models we compare against, our evaluation metrics, and implementation details, respectively.

### 2.4.1 Research questions

We seek to answer the following research questions:

(**RQ1.1**) Does MoGNet outperform state-of-the-art end-to-end single-module DRG models?

(**RQ1.2**) How does the choice of a particular collaboration mechanism (i.e., RMoG, PMoG, or neither of the two) affect the performance of MoGNet?

(**RQ1.3**) How does the GL learning scheme compare to using the general global learning as a learning scheme?

### 2.4.2 Dataset

Our experiments are conducted on the MultiWOZ [14] dataset. This is the latest large-scale human-to-human TDS dataset with rich semantic labels, e.g., domains and dialogue actions, and benchmark results of response generation.[2] MultiWOZ consists of $\sim$10k natural conversations between a tourist and a clerk. It has 6 specific action-related domains, i.e., *Attraction*, *Hotel*, *Restaurant*, *Taxi*, *Train*, and *Booking*, and one universal domain, i.e., *General*. 67.4% of the dialogues are cross-domain, with 2–5 domains on average. The average number of turns per dialogue is 13.68; a turn contains 13.18 tokens on average. The dataset is randomly split into 8,438/1,000/1,000 dialogues for training, validation, and testing, respectively.

### 2.4.3 Model variants and baselines

We consider a number of variants of the proposed mixture-of-generators model:

- **MoGNet**: the proposed model with RMoG and PMoG and GL learning scheme.
- **MoGNet-P**: the model without prospection ability by removing PMoG collaboration mechanism from MoGNet.

[2] http://dialogue.mi.eng.cam.ac.uk/index.php/corpus/

Table 2.1: Model variants. $\beta_j^C$, $\beta_j^{l,R}$, $\beta_j^{l,P}$ are from Eq. 2.5. "True" means we preserve it and learn it as it is. "False" means we remove it (set it to 0). $\lambda$ is from Eq. 2.11 and we report two settings, 0.0 and 0.5. See Section 2.6.2.

| | $\boldsymbol{\beta_j^C}$ | $\boldsymbol{\beta_j^{l,R}}$ | $\boldsymbol{\beta_j^{l,P}}$ | $\boldsymbol{\lambda}$ |
|---|---|---|---|---|
| MoGNet | True | True | True | 0.5 |
| MoGNet-P | True | True | False | 0.5 |
| MoGNet-P-R | True | False | False | 0.5 |
| MoGNet-GL | True | True | True | 0.0 |

- **MoGNet-P-R**: the model without the two collaboration policies but with GL learning scheme.
- **MoGNet-GL**: the model that removes GL learning scheme from MoGNet.

See Table 2.1 for a summary. Without further indications, the *intents* used are based on identifying eight different domains: Attraction, Booking, Hotel, Restaurant, Taxi, Train, General, and UNK.

To answer RQ1, we compare MoGNet with the following methods that have reported results on this task according to the official leaderboard by June 2020.[3]

- **S2SAttnLSTM**. We follow the dominant Seq2Seq model under an encoder-decoder architecture [20] and reproduce the benchmark baseline, i.e., the single-module model named S2SAttnLSTM [13, 14], based on the source code provided by the authors. See footnote 3.
- **S2SAttnGRU**. A variant of S2SAttnLSTM, with Gated Recurrent Units (GRUs) instead of LSTMs and other settings kept the same.
- **Structured Fusion**. It learns the traditional dialogue modules and then incorporates these pre-trained sequentially dependent modules into end-to-end dialogue models by structured fusion networks [142].
- **LaRLAttnGRU**. The state-of-the-art model [256], which uses reinforcement learning and models system actions as latent variables. LaRLAttnGRU uses ground truth system action annotations and user goals to estimate the rewards for reinforcement learning during training.

### 2.4.4 Evaluation metrics

We use the following commonly used evaluation metrics [14, 256]:

- *Inform*: the fraction of responses that provide a correct entity out of all responses.
- *Success*: the fraction of responses that answer all the requested attributes out of all responses.

[3] The Context-to-Text Generation task at `https://github.com/budzianowski/multiwoz`.

- *BLEU*: for comparing the overlap between a generated response to one or more reference responses.
- *Score*: defined as $Score = (0.5 * Inform + 0.5 * Success + BLEU) * 100$. This measures the overall performance in terms of both task completion and response fluency [142].
- *PPL*: denotes the perplexity of the generated responses, which is defined as the exponentiation of the entropy. This measures how well a probabilistic DRG model predicts a token in a response generation process.

We use the toolkit released by Budzianowski et al. [13] to compute the metrics.[4] Following their settings, we also use *Score* as the selection criterion to choose the best model on the validation set and report the performance of the model on the test set. We use a paired t-test to measure the statistical significance ($p < 0.01$) of relative improvements.

### 2.4.5 Implementation details

Theoretically, the training time complexity of each data sample is $\mathcal{O}(n * (k + 1) * n)$, where $n$ is the number of response tokens. To reduce the computational costs, we assign $j + t = n$ and compute the expert prediction with Eq. 2.3. This means that the chair will make a final decision only after all the experts have decoded their final tokens. Thus, the time complexity decreases to $\mathcal{O}(n * (k + 1) + n)$.

For a fair comparison, the vocabulary size is the same as used by Budzianowski et al. [14], who use 400 tokens. Out-of-vocabulary words are replaced with "[UNK]". We set the word embedding size to 50 and all GRU hidden state sizes to 150. We use Adam [91] as our optimization algorithm with hyperparameters $\alpha = 0.005$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. We also apply gradient clipping [162] with range [–5, 5] during training. We use $l2$ regularization to alleviate overfitting, the weight of which is set to $10^{-5}$. We set the mini-batch size to 64. We use greedy search to generate the responses during testing. Please note that if a data point has multiple intents, then we assign it to each corresponding expert, respectively. The code is available online.[5]

## 2.5 Results

In this section, we conduct both automatic and human evaluations to assess the effectiveness of the proposed model. We have also reported on an ablation study to show how collaboration policies and the learning scheme influence the proposed model.

### 2.5.1 Automatic evaluation (RQ1.1)

We evaluate the overall performance of MoGNet and the comparable baselines on the metrics defined in Section 2.4.3. The results are shown in Table 2.2. First of all,

[4] `https://github.com/budzianowski/multiwoz.`

[5] `https://github.com/Jiahuan-Pei/multiwoz-mdrg`

Table 2.2: Comparison results of MoGNet and the baselines. **Bold face** indicates leading results. Significant improvements over the best baseline are marked with * (paired t-test, $p < 0.01$).

| | **BLEU** | **Inform** | **Success** | **Score** | **PPL** |
|---|---|---|---|---|---|
| S2SAttnLSTM | 18.90% | 71.33% | 60.96% | 85.05 | **3.98** |
| S2SAttnGRU | 18.21% | 81.50% | 68.80% | 93.36 | 4.12 |
| Structured Fusion [142] | 16.34% | 82.70% | 72.10% | 93.74 | – |
| LaRLAttnGRU [256] | 12.80% | 82.78% | **79.20%** | 93.79 | 5.22 |
| MoGNet | **20.13%*** | **85.30%*** | 73.30% | **99.43*** | 4.25 |

MoGNet outperforms all baselines by a large margin in terms of overall performance metric, i.e., satisfaction *Score*. It significantly outperforms the state-of-the-art baseline LaRLAttnGRU by 5.64% (*Score*) and 0.97 (*PPL*). Thus, MoGNet not only improves the satisfaction of responses, but also improves the quality of the language modeling process. MoGNet also achieves more than 6.70% overall improvement over the benchmark baseline S2SAttnLSTM and its variant S2SAttnGRU. This proves the effectiveness of the proposed MoGNet model.

Second, LaRLAttnGRU achieves the highest performance in terms of *Success*, followed by MoGNet. However, it results in a 7.33% decrease in *BLEU* and a 2.56% decrease in *Inform* compared to MoGNet. Hence, LaRLAttnGRU is good at answering all requested attributes, but not as good at providing more appropriate entities with high fluency as MoGNet. LaRLAttnGRU tends to generate more slot values to increase the probability of answering the requested attributes. Take an extreme case as an example: if we force a model to generate all tokens with slot values, then it will achieve an extremely high *Success* but a low *BLEU*.

Third, S2SAttnLSTM is the worst performing model in terms of overall score (*Score*). But it achieves the best *PPL*. It tends to generate frequent tokens from the vocabulary, which exhibits better language modeling characteristics. However, it fails to provide useful information (the requested attributes) to meet the user goals. By contrast, MoGNet improves user satisfaction (i.e., *Score*) greatly and achieves response fluency by taking specialized generations from all experts into account.

### 2.5.2 Human evaluation (RQ1.1)

To further understand the results in Table 2.2, we conducted a human evaluation of the generated responses from S2SAttnGRU, LaRLAttnGRU, and MoGNet. We ask workers on Amazon Mechanical Turk (AMT)[6] to read the dialogue context, and choose the responses that satisfy the following criteria: (i) *Informativeness* measures whether the response provides appropriate information that is requested by the user query. No extra inappropriate information is provided. (ii) *Consistency* measures whether the generated response is semantically aligned with the ground truth response. (iii) *Satisfactory* measures whether the response has an overall satisfactory performance,

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

Table 2.3: Results of human evaluation. **Bold face** indicates the best results. $\geqslant n$ means that at least $n$ AMT workers regard it as a good response w.r.t. *Informativeness*, *Consistency* and *Satisfactory*.

| | S2SAttnGRU | | LaRLAttnGRU | | MoGNet | |
|---|---|---|---|---|---|---|
| | $\geqslant 1$ | $\geqslant 2$ | $\geqslant 1$ | $\geqslant 2$ | $\geqslant 1$ | $\geqslant 2$ |
| Informativeness | 56.79% | 31.03% | 76.54% | 44.83% | **80.25%** | **53.45%** |
| Consistency | 45.21% | 23.53% | 71.23% | 39.22% | **80.82%** | **50.98%** |
| Satisfactory | 26.79% | 25.00% | 44.64% | 21.88% | **60.71%** | **37.50%** |

promising both *Informativeness* and *Consistency*. As with existing studies [142], we sample one hundred context-response pairs to do the human evaluation. Each sample is labeled by three workers. The workers are asked to choose either all responses that satisfy the specific criteria or the "NONE" option, which denotes none of the responses satisfy the criteria. To make sure that the annotations are of high quality, we calculate the fraction of the responses that satisfy each criterion out of all responses that pass the *golden test*. That is, we only consider the data from the workers who have chosen the golden response as an answer.

The results are displayed in Table 2.3. MoGNet performs better than S2SAttnGRU and LaRLAttnGRU on *Informativeness* because it frequently outputs responses that provide richer information (compared with S2SAttnGRU) and fewer extra inappropriate information (compared with LaRLAttnGRU). MoGNet obtains the best results, which means MoGNet is able to generate responses that are semantically similar to the golden responses with large overlaps. The results of LaRLAttnGRU outperform S2SAttnGRU in all cases except for *Satisfactory* under the strict condition ($\geqslant 2$). This reveals that balancing between *Informativeness* and *Consistency* makes it difficult for the mturk workers to assess the overall quality, measured by *Satisfactory*. In this case, MoGNet receives the most votes on *Satisfactory* under the strict condition ($\geqslant 2$) as well as the loose condition ($\geqslant 1$). This shows that the workers consider the responses from MoGNet more appropriate than the other two models with a high degree of agreement. To sum up, MoGNet is able to generate user-favored responses in addition to the improvements for automatic metrics.

### 2.5.3 Collaboration policies (RQ1.2)

In Table 2.4 we contrast the effectiveness of different collaboration policies. We can see that MoGNet-P loses 4.32% overall performance with a 0.62% decrease of *BLEU*, 5.90% decrease of *Inform* and 1.50% decrease of *Success*. This shows that the prospective design of the PMoG mechanism is beneficial to both task completion and response fluency. Especially, most improvements come from providing more correct entities while improving generation fluency. MoGNet-P-R reduces 2.62% *Score* with 1.97% lower of *BLEU*, 0.2% lower of *Inform* and 1.10% of *Success*. Thus, the MoGNet framework is effective thanks to its design with two types of roles: the chair and the experts.

Table 2.4: Impact of collaboration policies. Underlined results indicate the worst results with a statistically significant decrease compared to MoGNet (paired t-test, $p < 0.01$).

| | **BLEU** | **Inform** | **Success** | **Score** | **PPL** |
|---|---|---|---|---|---|
| MoGNet | 20.13% | 85.30% | 73.30% | 99.43 | 4.25 |
| MoGNet-P | 19.51% | 79.40% | 71.80% | 95.11 | 4.19 |
| MoGNet-P-R | 18.16% | 85.10% | 72.20% | 96.81 | 4.12 |

Table 2.5: Impact of the learning scheme. Underlined results indicate the worst results with a statistically significant decrease compared with MoGNet (paired t-test, $p < 0.01$).

| | **BLEU** | **Inform** | **Success** | **Score** | **PPL** |
|---|---|---|---|---|---|
| MoGNet | 20.13% | 85.30% | 73.30% | 99.43 | 4.25 |
| MoGNet-GL | 19.33% | 78.40% | 67.90% | 92.48 | 3.97 |

### 2.5.4 Learning scheme (RQ1.3)

We use MoGNet-GL to refer to the model that removes the GL learning scheme from MoGNet and uses the general global learning instead. MoGNet-GL results in a sharp reduction of 6.95% overall performance with 0.80% of *BLEU*, 6.90% of *Inform* and 5.40% of *Success*. The main improvement is attributed to the strong task completion ability. This shows the effectiveness and importance of the GL learning scheme as it encourages each expert to specialize on a particular intent while the chair prompts all experts to collaborate with each other.

## 2.6 Analysis

In this section, we explore MoGNet in more detail. In particular, we examine (i) whether the intent partition affects the performance of MoGNet (Section 2.6.1); (ii) whether the improvements of MoGNet could simply be attributed to having a larger number of parameters (Section 2.6.2); (iii) how the hyper-parameter $\lambda$ (Eq. 2.11) affects the performance of MoGNet (Section 2.6.2); and (iv) how RMoG, PMoG and GL influence DRG using a small case study (Section 2.6.3).

### 2.6.1 Intent partition analysis

As stated above, the responses vary a lot for different intents which are differentiated by the domain and the type of system action. Therefore, we experiment with two types of intents as shown in Table 2.6.

To determine whether the intent partition affects the performance of MoGNet, we compared two ways of partitioning intents. MoGNet-domain and MoGNet-action denote the intent partitions w.r.t. domains and system actions, respectively. MoGNet-domain has 8 intents (domains) and MoGNet-action has 14 intents (actions), as shown in Table 2.6. The results are shown in Table 2.7.

Table 2.6: Two groups of intents that are divided by domains and the type of system actions.

| Type | Intents |
|---|---|
| Domain | Attraction, Booking, Hotel, Restaurant, Taxi, Train, General, UNK. |
| Action | Book, Inform, NoBook, NoOffer, OfferBook, OfferBooked, Select, Recommend, Request, Bye, Greet, Reqmore, Welcome, UNK. |

Table 2.7: Results of MoGNet with two intent partition ways.

| | BLEU | Inform | Success | Score | PPL |
|---|---|---|---|---|---|
| MoGNet-domain | **20.13%** | **85.30%** | **73.30%** | **99.43** | **4.25** |
| MoGNet-action | 17.28% | 79.40% | 69.70% | 91.83 | 4.48 |

MoGNet consistently outperforms the baseline S2SAttnGRU for both ways of partitioning intents. Interestingly, MoGNet-domain greatly outperforms MoGNet-action. We believe there are two reasons: First, the system actions are not suitable for grouping intents because some partition subsets are hard to be distinguished from each other, e.g., *OfferBook* and *OfferBooked*. Second, some system actions only have a few data samples, simply not enough to specialize the experts. The results show that different ways of partitioning intents may greatly affect the performance of MoGNet. Therefore, more effective intent partition methods, e.g., adaptive implicit intent partitions, need to be explored in future work.

### 2.6.2 Hyper-parameter analysis

To determine whether the improvements of MoGNet could simply be attributed to having a larger number of parameters, we show the results of MoGNet and S2SAttnGRU with different hidden sizes in Figure 2.3. S2SAttnGRU outperforms MoGNet when the number of parameters is less than 0.6e7. However, MoGNet achieves much better results with more parameters. Most importantly, the results from both models show that a larger number of parameters does not always mean better performance, which indicates that the improvement of MoGNet is not simply due to more parameters.

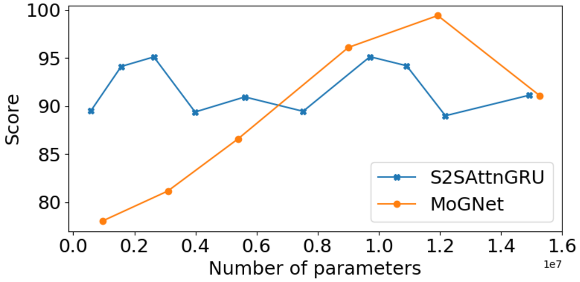


Figure 2.3: *Score* of MoGNet and S2SAttnGRU with different number of parameters.

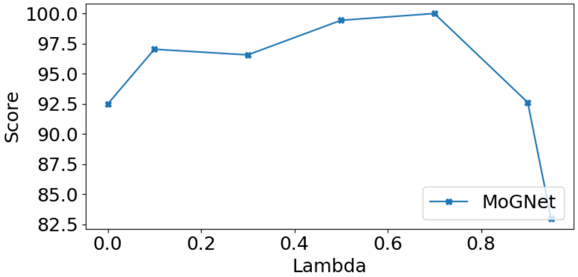


Figure 2.4: *Score* of MoGNet with different values of $\lambda$.

To understand how the hyper-parameter $\lambda$ affects the performance of MoGNet, we report the *Score* values of MoGNet with different values of $\lambda$ (Eq. 2.11), as shown in Figure 2.4. When $\lambda = 0$, no expert is trained on a particular intent. When $\lambda = 1$, the model ignores the global loss, i.e., the RMoG and PMoG mechanisms are not used and the chair is only trained as a general expert. We can see that these two settings decrease the performance greatly, which further verifies the effectiveness of RMoG and PMoG as well as the MoGNet framework. We also note that the performance of MoGNet is quite stable when $\lambda \in [0.1, 0.7]$ with the best performance for $\lambda = 0.7$. Hence, MoGNet is not very sensitive to the hyper-parameter $\lambda$.

### 2.6.3 Case study

To address how RMoG, PMoG and GL influence DRG, we select an example to illustrate the influence of RMoG, PMoG, and GL. Table 2.8 shows the responses generated by comparable baselines (i.e., S2SAttnGRU, LaRLAttnGRU) and MoGNet variants as in Table 2.4. In red we highlight the tokens that show the differences in terms of task completion. Generally, MoGNet can generate more appropriate and meaningful responses. Specifically, without PMoG, MoGNet-P and MoGNet-P-R ignore the fact that the attribute *time* is important for searching a train ticket (1st turn) and omit the exact departure time ([value_time]) of the train (2nd turn). Without GL, MoGNet-GL ignores the primary time information need *day* (1st turn) and omits the implicit need of [value_price] (4th turn).

There are also some low-quality cases, e.g., MoGNet and the baselines occasionally generate redundant and lengthy responses, because none of them has addressed this issue explicitly during training.

## 2.7 Conclusion and future work

In this chapter, we have proposed a novel mixture-of-generators network (MoGNet) model with different collaboration policies, namely, RMoG and PMoG, to enhance dialogue response generation. We have also devised a GL learning scheme to learn MoGNet effectively. Experiments on the MultiWOZ benchmark demonstrate that MoGNet significantly outperforms state-of-the-art methods in terms of both automatic and human evaluations. We also conduct analyses that confirm the effectiveness of MoGNet, the RMoG and PMoG mechanisms, as well as the GL learning scheme.

Table 2.8: Example responses from MoGNet with the ablation settings in a 4-turn dialogue.

| Turn | User | System |
|---|---|---|
| 1 | i am looking for a train departing from [value_place] that will arrive at [value_place]. | **S2SAttnGRU**: what day would you like to travel ?<br>**LaRLAttnGRU**: there are [value_count] trains leaving [value_place] on [value_day] . what day would you like to travel ?<br>**MoGNet**: what day and time would you like to travel?<br>**MoGNet-P**: what day will you be traveling?<br>**MoGNet-P-R**: what day will you be traveling?<br>**MoGNet-GL**: there are [value_count] trains ... do you have a time you would like to depart? |
| 2 | i would like to leave on [value_day] and arrive no later than [value_time]. | **S2SAttnGRU**: [train_id] will arrive at [value_time] .<br>**LaRLAttnGRU**: i have train [train_id] leaving at [value_time] and arriving at [value_time] . would you like me to book this for you ?<br>**MoGNet**: [train_id] leaves at [value_time] and arrives by [value_time] .<br>**MoGNet-P**: [train_id] will arrive at [value_time].<br>**MoGNet-P-R**: the [train_id] arrives at [value_time].<br>**MoGNet-GL**: i have train [train_id] that leaves at [value_time] and arrives at [value_time].<br>+*All*: would you like me to book that for you? |
| 3 | can i please have the reference number? | *All*: ... how many tickets do you need? |
| 4 | could you book [value_count] tickets for me and send me the reference number? | **S2SAttnGRU**: i have booked you [value_count] tickets on the [train_id] . your reference number is [train_reference]. is there anything else i can help you with ?<br>**LaRLAttnGRU**: i have booked [value_count] tickets on [train_id] . the reference number is [train_reference] .<br>**MoGNet**: i have booked [value_count] tickets for you. the total fee is [value_price] gbp payable at the station. your reference number is [train_reference]. is there anything else i can help you with ?<br>**MoGNet-P**: i have booked [value_count] tickets. the total fee is [value_price] gbp payable at the station. your reference number is [train_reference].<br>**MoGNet-P-R**: booking was successful , the total fee is [value_price] gbp payable at the station. reference number is [train_reference].<br>**MoGNet-GL**: i have booked [value_count] tickets for you. the reference number is [train_reference]. is there anything else i can help you with ? |

Together, these findings constitute a positive answer to our leading research question for this chapter: multiple dialogue agents can indeed be organized in such a way that they collaborate effectively to improve the performance of a single-module agent.

As to future work that builds on the findings of this chapter, we plan to devise more fine-grained expert generators and experiment on more datasets to test MoGNet. In addition, MoGNet can be advanced in many directions: First, better mechanisms can be proposed to improve the collaboration between the chair and expert generators. Second, it would be interesting to study how to do intent partition automatically. Third, it is also important to investigate how to avoid redundant and lengthy responses in order to provide a better user experience.

Next, we switch from model collaboration to user collaboration.

# 3

# User Collaboration: A Cooperative Memory Network for Personalization

In this chapter, we aim to answer the following research question:

**RQ2** Can multiple users collaborate successfully to improve the quality of a dialogue for each single user?

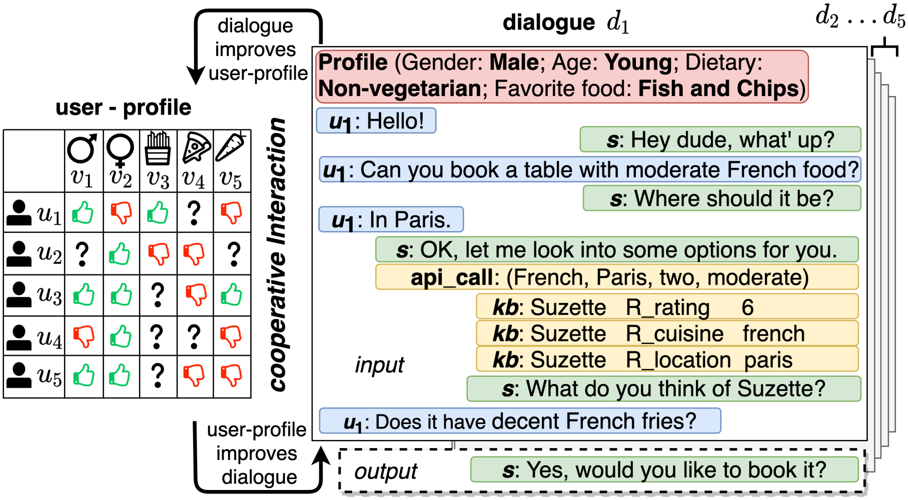
We examine the idea of collaborative agents from the point of view of user collaboration. We propose a cooperative memory network (CoMemNN) that can gradually enrich user profiles with collaborative users as dialogues progress and simultaneously improve the performance of response selection based on the enriched profiles. Our main finding is that CoMemNN enriches user profiles effectively with robustness, leading to a continuous increase of accuracy for response selection compared to state-of-the-art baselines.

## 3.1 Introduction

The use of task-oriented dialogue systems (TDSs) is becoming increasingly widespread. Unlike open-ended dialogue systems [106, 252], TDSs are meant to help users achieve specific goals during multiple-turn dialogues [20]. Applications include booking restaurants, planning trips, grocery shopping, customer service [e.g., 12, 133, 166, 168, 232, 249].

Considerable progress has been made in improving the performance of TDSs [e.g., 12, 71, 111, 116, 124, 167, 234]. Human-human dialogues naturally reflect diverse personalized preferences in terms of, e.g., modes of expression habits [48, 250], individual needs and related to specific goals [85, 126, 145]. Recent work has begun to explore how to improve the user experience by personalizing TDSs in similar ways. Several personalized TDS models have been proposed and have achieved good performance [85, 126, 251]. Personalized TDS models use user profiles in order to be able to capture, and optimize for, users' personal preferences. Those user profiles may not

This chapter was published as J. Pei, P. Ren, and M. de Rijke. A cooperative memory network for personalized task-oriented dialogue systems with incomplete user profiles. In *The Web Conference*, pages 1552–1561, 2021 [169].



|  | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|---|---|---|---|---|---|
| $u_1$ | 👍 | 👎 | 👍 | ? | 👎 |
| $u_2$ | ? | 👍 | 👎 | 👎 | ? |
| $u_3$ | 👍 | 👍 | ? | 👎 | 👍 |
| $u_4$ | 👎 | 👍 | ? | ? | 👎 |
| $u_5$ | 👍 | 👍 | ? | 👎 | 👎 |

**Profile** (Gender: **Male**; Age: **Young**; Dietary: **Non-vegetarian**; Favorite food: **Fish and Chips**)

***u*1**: Hello!

***s***: Hey dude, what' up?

***u*1**: Can you book a table with moderate French food?

***s***: Where should it be?

***u*1**: In Paris.

***s***: OK, let me look into some options for you.

**api_call**: (French, Paris, two, moderate)

***kb***: Suzette R_rating 6

***kb***: Suzette R_cuisine french

***kb***: Suzette R_location paris

***s***: What do you think of Suzette?

***u*1**: Does it have decent French fries?

***s***: Yes, would you like to book it?

Figure 3.1: Cooperative interaction between user profiles and dialogues.

always be available or complete. While profiles may be obtained by asking users to fill in personal profiles with all predefined attributes [85, 126, 251], more often than not, they are *incomplete* and have missing values for some of the attributes of interest: (i) Not all users are willing to expose their profiles due to privacy concerns [224]; Tigunova et al. [212] have shown that users rarely reveal their personal information in dialogues explicitly; and (ii) User profiles may involve many attributes (e.g., gender, age, and tastes), which makes it hard to collect values for all of them. For example, even if we know a user's favorite food is "fish and chips," this does not mean the user does not like "hamburgers."

In this chapter, we study the problem of personalized TDSs with *incomplete user profiles*. This problem comes with two key challenges: (i) how to infer missing attribute values of incomplete user profiles; and (ii) how to use enriched profiles so as to enhance personalized TDSs. There have been previous attempts to extract user profiles from open-ended dialogues [104, 108, 211, 212, 236] but to the best of our knowledge, the problem of inferring and using missing attribute values has not been studied yet in the context of TDSs.

We address the problem of personalized TDSs with *incomplete user profiles* by proposing an end-to-end *cooperative memory network* (CoMemNN) in which profiles and dialogues are used to mutually improve each other. See Figure 3.1 for an intuitive sketch. The intuition behind CoMemNN is that user profiles can be gradually improved (i.e., missing values can be added) by leveraging useful information from each dialogue turn, and simultaneously, the performance of dialogue response selection (DRS) can be improved based on enriched profiles for later turns. For example, when user $u_1$ produces the utterance "Does it have 'decent' french fries?" and the user reveals his like of "french fries," the attribute 'favorite food' in his user profile can be enriched with the value of "french fries." In addition, we want to consider collaborative information from similar users, assuming that similar users have similar preferences as reflected in their user profiles. For example, a young male non-vegetarian who is a big fan of "pizza" might also love "fish and chips" if there are several users with similar profiles stating "fish and chips" as their favorite food. In turn, knowledge of these preferences can affect the choice of the response selected by a TDS in case there are multiple candidate responses. In other words, users with similar profiles may expect the same or a similar response given a certain dialogue context [126].

CoMemNN operationalizes the intuitions spelled out above with two key modules: *user profile enrichment* (UPE) and *dialogue response selection* (DRS). The former enriches incomplete user profiles by utilizing useful information from the current dialogue as well as collaborative information from similar users. The latter uses the enriched profiles to update the query representing all requested information, based on which a personalized response is selected to reply to user requests.

To verify the effectiveness of CoMemNN, we conduct extensive experiments on the personalized bAbI dialogue (PbAbI) benchmark dataset, which comes in two flavors, a small version that has 1,000 dialogues, and a large version, which has 12,000 dialogues. First, we find that CoMemNN improves over the best baseline by 3.06%/2.80% on the small/large dataset, respectively, when using all available user profiles. Second, to assess the performance of CoMemNN in the presence of incomplete user profiles, we randomly discard values of attributes with varying probabilities and find that even when we discard 50% of the attribute values, the performance of CoMemNN matches the performance of the best performing baseline without discarding user profiles. In contrast, the best performing baseline decreases by 2.12%/1.97% in performance on the small/large dataset with the same amount of discarded values.

The main contributions of this chapter are as follows:

- We consider the task of personalized TDSs with incomplete user profiles, which has not been investigated so far, to the best of our knowledge.
- We devise a CoMemNN model with dedicated modules to gradually enrich user profiles as a dialogue progresses and to improve response selection based on enriched profiles at the same time.
- We carry out extensive experiments to show the robustness of CoMemNN in the presence of incomplete user profiles.

## 3.2 Related work

In this section, we present a brief overview of related work on personalized open-ended dialogue systems and personalized task-oriented dialogue systems (TDSs).

### 3.2.1 Personalized open-ended dialogue systems

Previous studies on personalized open-ended dialogue systems mainly fuse unstructured persona information [138, 252]. Li et al. [106] first attempt to incorporate a persona into the Seq2Seq framework [207] to generate personalized responses. Ficler and Goldberg [48] apply an RNN language model conditioned on a persona to control response generation with linguistic style.

Zhang et al. [252] find that selection models based on Memory Networks [206] are more promising than recurrent generation models based on Seq2Seq [207]. Mazare et al. [138] develop a response selection model based on MemNN and model persona to improve the performance of an open-ended dialogue system. Song et al. [200] explore how to generate diverse personalized responses using a variational autoencoder conditioned on a persona memory. Liu et al. [117] make use of persona interaction between two interlocutors. Xu et al. [239] further exploit topical information to extend persona.

Prior attempts to address data sparsity problems in order to enhance personalized open-ended dialogue systems have considered pretraining [73, 259], sketch generation and filling [197], multiple-stage decoding [201], multi-task learning [125], transfer learning [233, 245, 253], and meta-learning [134]. Only few studies have explored structured user profiles for open-ended dialogue systems [173, 258, 262].

Most of the methods listed above focus on unstructured persona information while we target structured user profiles. Importantly, previous methods focus on open-ended dialogue systems, so they cannot be applied to TDSs directly.

### 3.2.2 Personalized task-oriented dialogue systems

Unlike open-ended dialogue systems, personalized TDSs have not been investigated extensively so far. Joshi et al. [85] release the first and, so far, only benchmark dataset for personalized TDSs, to the best of our knowledge. They propose a memory network based model, MemNN, to encode user profiles and conduct personalized response selection. They also propose an extension of MemNN, Split MemNN, which splits a memory into a profile memory followed by a dialogue memory.

Zhang et al. [251] introduce Retrieval MemNN by incorporating a retrieval module into memory networks, which enhances the performance by retrieving the relevant responses from other users. Luo et al. [126] present Personalized MemNN which learns distributed embeddings for user profiles, dialogue history, and the dialogue history from users with the same gender and age, and shows better performance by using the idea of user bias towards knowledge base (KB) entries over candidate responses. Mo et al. [145] introduce a transfer reinforce learning paradigm to alleviate data scarcity, which uses a collection of multiple users as a source domain and an individual user as a target domain.

The methods mentioned above all assume that complete user profiles can be obtained by urging users to fill in all blanks in user profiles, which is unrealistic in practice. Thus, it remains unexplored how the methods above perform when incomplete user profiles are provided, and whether we can bridge the gap in performance if their performance is negatively affected. An alternative would be to first infer missing user profiles, e.g., by mining query logs or previous conversations [104, 108, 211, 212], and then apply the model with the above methods. But to do so, we would need to train a model to infer missing user profiles asynchronously. Besides, it would likely bring cumulative errors to downstream TDS tasks.

Instead, we propose to enrich user profiles *and* address the core TDS task simultaneously with an end-to-end model.

Table 3.1: Summary of main notation used in the chapter.

| | |
|---|---|
| $X_t^u$ | User utterance at turn $t$. |
| $X_t^s$ | System response at turn $t$. |
| $D_t$ | Dialogue history at turn $t$. |
| $\mathbf{h}_t$ | Hidden representation of $D_t$. |
| $u$ | A user profile in the form of $\{(a_i, v_i)\}_{i=1}^m$, $v_i$ is a candidate value of $i$-th attribute $a_i$. |
| $\mathbf{p}$ | One-hot representation of $u$. |
| $\mathbf{q}_t$ | A query representation that represents the user's current request at turn $t$. |
| $\mathbf{M}_t^P$ | Profile memory that contains user profile presentations of $u$ and his/her neighbors at turn $t$. |
| $\mathbf{M}_t^D$ | Dialogue memory that contains dialogue history presentation of $u$ and his/her neighbors at turn $t$. |

## 3.3 Method

### 3.3.1 Task

In this work, we follow previous studies and model a personalized TDS as a response selection task, which selects a response from predefined candidates given a dialogue context [46, 85, 126, 167, 178, 229, 251]. Table 3.1 summarizes the main notation used in this chapter.

Given a dialogue context $(u, D_t, X_t^u)$ at the $t$-th dialogue turn, our goal is to select an appropriate response $y_t = X_t^s$ from candidate responses $Y = \{X_j^s\}|_{j=1}^{|Y|}$. Here, $u$ is the user profile, which consists of $m$ attribute-value pairs $\{(a_i, v_i)\}_{i=1}^m$, where $a_i$ is the $i$-th attribute and $v_i$ is a candidate value of $a_i$. For example, in Figure 3.1, the user profile is denoted as $\{$(Gender, Male), (Age, Young), (Dietary, Non-vegetarian), (Favorite food, Fish and Chips)$\}$. $D_t = X_{1:t-1}$ is the dialogue history. Similar to [85, 126, 251], $D_t$ is represented as a sequence of words that are aggregated from historical utterances $[X_1^u, X_1^s, \ldots, X_{t-1}^u, X_{t-1}^s]$, alternating between the user $u$ or system $s$. $X_t^u$ denotes the current user utterance, representing the user's current request.

### 3.3.2 Overview of CoMemNN

A high-level overview of the proposed architecture, CoMemNN, is shown in Figure 3.2, while a more detailed view of the pipeline is offered in Figure 3.3. A key aspect of the architecture is that it aims to capture all useful information from the given dialogue context $(u, D_t, X_t^u)$, based on which we learn a query representation $\mathbf{q}_t$ to represent the user's current request. $\mathbf{q}_t$ is usually initialized with the current user utterance $X_t^u$ [85, 126, 251]. Then, $\mathbf{q}_t$ is updated by the user profile enrichment (UPE) module by incorporating dialogue and personal information from dialogues and user profiles, respectively. Specifically, UPE captures the interaction between user profiles and dialogues with three submodules: memory initialization (MI), memory updating (MU)

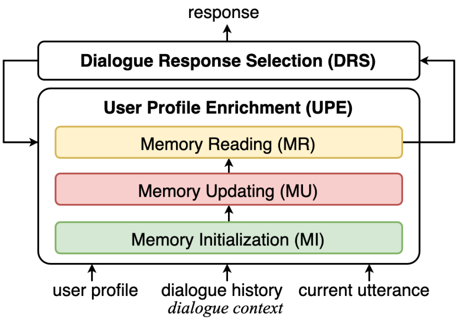


Figure 3.2: An overview of the CoMemNN architecture, which consists of two cooperative modules: user profile enrichment (UPE) and dialogue response selection (DRS).

and memory reading (MR). MI searches neighbors of the current user to initialize the profile memory $\mathbf{M}_t^P$, which contains profiles from both the current user and his/her neighbors. MI also initializes the dialogue memory $\mathbf{M}_t^D$ with the dialogue history of both the current user and his/her neighbors, each of which is represented by addressing dialogue historical utterance representations with $\mathbf{q}_t$. MU updates the profile memory $\mathbf{M}_t^P$ and the dialogue memory $\mathbf{M}_t^D$ by considering their interaction, after which the user profiles are enriched by inferring missing values based on the dialogue and personal information from the current user and his/her neighbors. After this, MR updates the query representation $\mathbf{q}_t$ by reading from the enriched profile memory as well as dialogue memory. Finally, the dialogue response selection (DRS) module uses the updated query to match candidate responses so as to select an appropriate response. Next, we introduce each of the modules MI, MU, and MR, one by one.

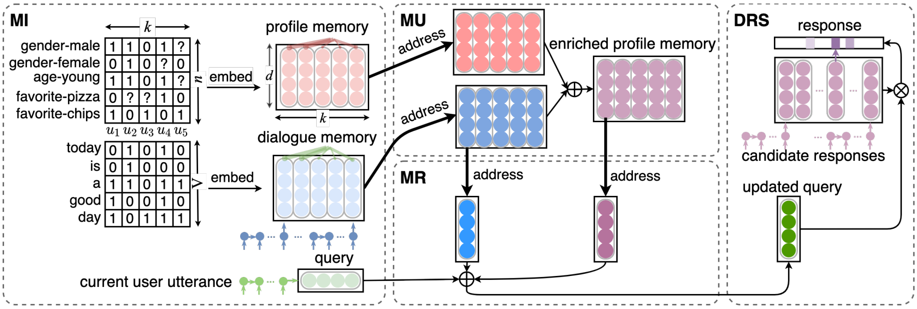
### 3.3.3 Memory initialization (MI)

#### Profile memory initialization

To model user-profile relations, we initialize the profile memory as: $\mathbf{M}_t^P = [\Psi(u_1), \ldots, \Psi(u_k)] \in \mathbb{R}^{k \times d}$, where $u_1$ is the current profile (CP) from the current user. The others are neighbor profiles (NPs) from neighbor users. For each user profile, the $i$-th attribute can be represented as an one-hot vector $\tilde{\mathbf{p}}_i \in \mathbb{R}^{C(p_i)}$, where there are $C(a_i)$ candidate values for $p_i$. Then, each user profile can be initialized as an one-hot vector $\mathbf{p} = \text{Concat}(\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_m) \in \mathbb{R}^n (n = \sum_{i=1}^m (C(p_i))$, which is the concatenation of one-hot representations of attributes. $k$ is the number of users, $d$ is the embedding dimension, and $\Psi$ is a linear transformation function. Given any user profile $u$, we find his/her $(k-1)$ nearest neighbors based on dot product similarity.

#### Dialogue memory initialization

To model user-dialogue relations, we initialize a dialogue memory $\mathbf{M}_t^D = [\mathbf{h}_t^1, \ldots, \mathbf{h}_t^k] \in \mathbb{R}^{k \times d}$, where $\mathbf{h}_t^1$ is the representation of the current dialogue (CD) from the current user. The others are the neighbor dialogues (NDs) from neighbor users. For each user, the



| | | | | | |
|---|---|---|---|---|---|
| gender-male | 1 | 1 | 0 | 1 | ? |
| gender-female | 0 | 1 | 0 | ? | 0 |
| age-young | 1 | 1 | 0 | 1 | ? |
| favorite-pizza | 0 | ? | ? | 1 | 0 |
| favorite-chips | 1 | 0 | 1 | 0 | 1 |
| | $u_1$ | $u_2$ | $u_3$ | $u_4$ | $u_5$ |

| | | | | | |
|---|---|---|---|---|---|
| today | 0 | 1 | 0 | 1 | 0 |
| is | 0 | 1 | 0 | 0 | 0 |
| a | 1 | 1 | 0 | 1 | 1 |
| good | 1 | 0 | 0 | 1 | 0 |
| day | 1 | 0 | 1 | 1 | 1 |

Figure 3.3: A more detailed view of the pipeline of the CoMemNN model. The UPE modules captures the interaction between user profiles and dialogues by three submodules: MI, MU and MR. The DRS module and the UPE module cooperate so as to select better responses. Section 3.3 contains a walkthrough of the model.

dialogue history can be computed as:

$$
\begin{aligned}
\mathbf{h}_t &= \sum_{i=1}^{2(t-1)} \lambda_t^i \mathbf{H}_t^i \in \mathbb{R}^d \\
\lambda_t^i &= (\tilde{\mathbf{q}}_t)^T \cdot \mathbf{H}_t^i \in \mathbb{R}^1,
\end{aligned}
\tag{3.1}
$$

where we use the updated query $\tilde{\mathbf{q}}_t$ to address the aggregated dialogue history $\mathbf{H}_t$, the addressing weight $\lambda_t^i$ is computed by the dot product of query $\tilde{\mathbf{q}}_t$ and the $i$-th utterance representation $\mathbf{H}_t^i$.

Following [43, 126], we represent each utterance as a bag-of-words using the embedding matrix $\mathbf{E} \in \mathbb{R}^{d \times V}$, where $d$ is the embedding dimension, $V$ is the vocabulary size, $\Phi(\cdot)$ maps the utterance to a bag of dimension $V$. At the beginning of turn $t$, the updated query $\tilde{\mathbf{q}}_t$ is initialized as:

$$
\tilde{\mathbf{q}}_t = \mathbf{E}\Phi(X_t^u) \in \mathbb{R}^d. \tag{3.2}
$$

Similarly, the aggregated dialogue history $\mathbf{H}_t$ of the current user $u_1$ can be embedded as:

$$
\mathbf{H}_t = [\mathbf{E}\Phi(X_1^u), \mathbf{E}\Phi(X_1^s), \ldots, \mathbf{E}\Phi(X_{t-1}^u), \mathbf{E}\Phi(X_{t-1}^s)] \in \mathbb{R}^{2(t-1) \times d}. \tag{3.3}
$$

## 3.3.4 Memory updating (MU)

### Dialogue memory updating

To obtain the intermediate dialogue memory $\tilde{\mathbf{M}}_t^D$, we update the $i$-th dialogue memory slot $\tilde{\mathbf{M}}_t^D[:, i]$ using the newest updated query $\tilde{\mathbf{q}}_t$ to address initial dialogue memory

$\mathbf{M}_t^D$ as:

$$\begin{aligned}\tilde{\mathbf{M}}_t^D[:,i] &= \sum_{j=1}^{k} \beta_t^j \mathbf{M}_t^D[:,j] \in \mathbb{R}^d \\ \beta_t^j &= (\tilde{\mathbf{q}}_t)^T \cdot \mathbf{M}_t^D[:,j] \in \mathbb{R}^1.\end{aligned} \tag{3.4}$$

Next, the initial dialogue memory is updated by assigning $\mathbf{M}_t^D = \tilde{\mathbf{M}}_t^D$. As the dialogue evolves, the profile memory will gradually improve the dialogue memory because $\tilde{\mathbf{q}}_t$ contains information from the previous profile memory, so addressing with $\tilde{\mathbf{q}}_t$ links profile-dialogue relations to the dialogue memory.

### Profile memory updating

Similarly, we can obtain an intermediate profile memory $\tilde{\mathbf{M}}_t^P$ with the following steps:

$$\begin{aligned}\tilde{\mathbf{M}}_t^P[:,i] &= \sum_{j=1}^{k} \alpha_t^j \mathbf{M}_t^P[:,j] \in \mathbb{R}^d \\ \alpha_t^j &= (\mathbf{M}_t^P[:,i])^T \cdot \mathbf{M}_t^P[:,j] \in \mathbb{R}^1.\end{aligned} \tag{3.5}$$

Next, the profile memory slot $\mathbf{M}_t^P[:,i]$ is updated by a function $\Gamma(\cdot)$ using the intermediate profile memory slot $\tilde{\mathbf{M}}_t^P[:,i]$ and the newest updated dialogue memory slot $\tilde{\mathbf{M}}_t^D[:,i]$:

$$\mathbf{M}_t^P[:,i] = \Gamma(\tilde{\mathbf{M}}_t^P[:,i], \tilde{\mathbf{M}}_t^D[:,i]) \in \mathbb{R}^d, \tag{3.6}$$

where $\Gamma(\cdot)$ is a mapping function that is implemented by a multiple layer perceptron (MLP) in this work. In this process, the dialogue memory helps to improve the profile memory because $\Gamma(\cdot)$ links dialogue-profile relations to the profile memory.

## 3.3.5 Memory reading (MR)

### Dialogue memory reading

Since the first memory slot corresponds to the current user, we compute $\mathbf{m}_t^D$ by hard addressing and use it to update the query $\tilde{\mathbf{q}}_t$ as follows:

$$\begin{aligned}\tilde{\mathbf{q}}_t &= \tilde{\mathbf{q}}_t + \mathbf{m}_t^D \in \mathbb{R}^d \\ \mathbf{m}_t^D &= \tilde{\mathbf{M}}_t^D[:,1] \in \mathbb{R}^d.\end{aligned} \tag{3.7}$$

### Profile memory reading

Similarly, we obtain $\mathbf{m}_t^P$ by hard addressing and use it to update the query $\tilde{\mathbf{q}}_t$ as follows:

$$\begin{aligned}\tilde{\mathbf{q}}_t &= \tilde{\mathbf{q}}_t + \mathbf{m}_t^P \in \mathbb{R}^d \\ \mathbf{m}_t^P &= \mathbf{M}_t^P[:,1] \in \mathbb{R}^d.\end{aligned} \tag{3.8}$$

### 3.3.6 Dialogue response selection

We use the latest updated query $\tilde{\mathbf{q}}_t$ to match with candidate dialogue responses and the predicted response distribution is computed as follows:

$$\begin{aligned}
\tilde{\mathbf{y}}_t &= \mathrm{softmax}(\tilde{\mathbf{q}}_t^T \mathbf{r}_1 + \mathbf{b}_1, \ldots, \tilde{\mathbf{q}}_t^T \mathbf{r}_{|Y|} + \mathbf{b}_{|Y|}) \in \mathbb{R}^{|Y|} \\
\mathbf{b}_j &= \begin{cases} \mathbf{f}_i \in \mathbb{R}^1 & \text{if } \mathbf{r}_j \text{ mentions } i\text{-th attribute of a KB entry} \\ 0 & \text{otherwise} \end{cases} \\
\mathbf{f} &= \mathrm{ReLU}(\mathbf{F}\mathbf{p}_1) \in \mathbb{R}^{kb},
\end{aligned} \tag{3.9}$$

where $\mathbf{r}_j$ is the representation of the $j$-th candidate response, $|Y|$ is the number of all candidate responses. We follow Luo et al. [126] to model the user bias towards KB entries over the $j$-th candidate response by a term $\mathbf{b}_j$, where the dimension $kb$ is the number of attributes of a KB entry. $\mathbf{p}_1 \in \mathbb{R}^n$ is the one-hot representation of the current user profile. $\mathbf{F} \in \mathbb{R}^{kb \times n}$ maps user profiles into a KB entry.

### 3.3.7 Learning of CoMemNN

Multiple-hop reading or updating has been shown to help improve performance of MemNN by reading or updating the memory multiple times [85, 126, 206]. To enhance CoMemNN, we devise a learning algorithm to update the query and memories with multiple hops, and further differentiate the specific losses of the UPE and DRS modules. The learning procedure is shown in Algorithm 1. First, MI searches neighbors $\{u_2, \ldots, u_k\}$ of the current user $u_1$ to initialize the profile memory $\mathbf{M}_t^P$ and dialogue memory $\mathbf{M}_t^D$. Second, MU and MR are conducted $HopN$ times, and for each time MU updates the dialogue memory $\mathbf{M}_t^D$ and the profile memory $\mathbf{M}_t^P$ by considering their cooperative interaction. After that, MR updates the query representation $\mathbf{q}_t$ by reading from the enriched dialogue memory followed by profile memory. Last, the dialogue response selection (DRS) module uses the newest updated query $\tilde{\mathbf{q}}_t$ to match candidate responses so as to predict a response distribution $\tilde{\mathbf{y}}_t$.

To evaluate the performance of DRS and UPE, we define two mapping functions to get prediction labels:

- $\mathrm{Argmax}(\cdot)$: it outputs the index $y_t$ with the highest probability in a predicted response distribution $\tilde{\mathbf{y}}_t$;
- $\mathrm{PiecewiseArgmax}(\cdot)$: it generates a 1-0 vector from the predicted enriched profile $\mathbf{m}_t^P$, where $\tilde{\mathbf{p}}_t^1[i] = 1$ only if $\mathbf{m}_t^P[i]$ achieves the highest probability among the values that belong to the same attribute.

To optimize DRS, we use a standard cross-entropy loss between the prediction $\tilde{\mathbf{y}}$ and the one-hot encoded true label $\mathbf{y}$:

$$\mathcal{L}_{\text{DRS}}(\theta) = -\frac{1}{N_1} \sum_{i=1}^{N_1} \sum_{j=1}^{|Y|} \mathbf{y}_j \log \tilde{\mathbf{y}}_j, \tag{3.10}$$

where $\theta$ are all parameters in the model and $N_1$ is the number of training samples.

**Algorithm 1:** Multiple hop CoMemNN.

```
Input: turn $t$, user $u_1$, profile $\mathbf{p}_1$, dialogue history $\mathbf{H}_t$, query $\mathbf{q}_t$, response
       candidates $\{\mathbf{r}_1, \ldots, \mathbf{r}_{|Y|}\}$, max hop $HopN$, $(k-1)$ neighbors
Output: A index $\mathbf{y}_t$ of next response; An one-hot vector $\tilde{\mathbf{p}}_t^1$ presenting the
        enriched profile.
$\{u_2, \ldots, u_k\} \leftarrow \mathrm{Search}(\mathbf{p}_1, k-1)$;                                  ▷ MI
$\mathbf{M}_t^P \leftarrow [\mathbf{p}_1, \ldots, \mathbf{p}_k]$;
$\mathbf{M}_t^D \leftarrow [\mathbf{h}_t^1, \ldots, \mathbf{h}_t^k]; \mathbf{h}_t^i \leftarrow (\tilde{q}_t, \mathbf{H}_t^i), i \in [1, k]; \tilde{\mathbf{q}}_t \leftarrow \mathbf{q}_t$;
while $hop \leq HopN$ do
    $\tilde{\mathbf{M}}_t^D \leftarrow \mathbf{M}_t^D; \tilde{\mathbf{M}}_t^P \leftarrow \mathbf{M}_t^P$ ;                                  ▷ MU
    $\mathbf{M}_t^D \leftarrow \tilde{\mathbf{M}}_t^D$;
    $\mathbf{M}_t^P \leftarrow \Gamma(\tilde{\mathbf{M}}_t^P, \tilde{\mathbf{M}}_t^D)$;
    $\mathbf{m}_t^D \leftarrow \mathbf{M}_t^D; \tilde{\mathbf{q}}_t \leftarrow \tilde{\mathbf{q}}_t + \mathbf{m}_t^D$ ;                                  ▷ MR
    $\mathbf{m}_t^P \leftarrow \mathbf{M}_t^P; \tilde{\mathbf{q}} \leftarrow \tilde{\mathbf{q}}_t + \mathbf{m}_t^P$;
$\tilde{\mathbf{y}}_t \leftarrow \mathrm{softmax}(\tilde{\mathbf{q}}_t^T \mathbf{r}_1 + \mathbf{b}_1, \ldots, \tilde{\mathbf{q}}_t^T \mathbf{r}_{|Y|} + \mathbf{b}_{|Y|})$ ;                ▷ DRS
$y_t \leftarrow \mathrm{Argmax}_j(\tilde{\mathbf{y}}_t)$;
$\tilde{\mathbf{p}}_t^1 \leftarrow \mathrm{PiecewiseArgmax}(\mathbf{m}_t^P)$
```

To control the learning of UPE, we introduce the element-wise mean squared loss between the sampled profile $\mathbf{p} = \{p_1, \ldots, p_{N_2}\}$ and its corresponding enriched profile $\tilde{\mathbf{p}} = \{\tilde{p}_1, \ldots, \tilde{p}_{N_2}\}$:

$$\mathcal{L}_{\text{UPE}}(\theta) = -\frac{1}{N_2} \sum_{i=1}^{N_2} (p_i - \tilde{p}_i), \tag{3.11}$$

where $\theta$ are all parameters in the model and $N_2$ is the number of sampled values.

Finally, the final loss is a linear combination:

$$\mathcal{L}(\theta) = \mu \mathcal{L}_{\text{DRS}}(\theta) + (1 - \mu) \mathcal{L}_{\text{UPE}}(\theta), \tag{3.12}$$

where $\mu$ is a hyper-parameter to balance the relative importance of the constituent losses.

# 3.4 Experimental setup

## 3.4.1 Research questions

We seek to answer the following questions in our experiments:

(**RQ2.1**) How well does CoMemNN perform? Does it significantly and continuously outperform state-of-the-art methods?

(**RQ2.2**) What are the effects of different components in CoMemNN?

(**RQ2.3**) Do different profile attributes contribute differently? and

(**RQ2.4**) How well does CoMemNN perform in terms of robustness?

### 3.4.2 Dataset and evaluation

We use the personalized bAbI dialogue (PbAbI) dataset [85] for our experiments; this is an extension of the bAbI dialogue (bAbI) dataset that incorporates personalization [12]. To the best of our knowledge, this is the only available open dataset for personalized TDSs. There are two versions: a large version with around 12,000 dialogues and a small version with 1,000 dialogues. These two datasets share the same vocabulary with 14,819 tokens and a candidate response set with 43,863 responses.

The dataset defines four user profile attributes (gender, age, dietary preference, and favorite food) and composes corresponding attribute-value pairs to a user profile. Each conversation is provided with all of the above user profile attributes, e.g., {(Gender, Male), (Age, Young), (Dietary, Non-vegetarian), (Favorite: Fish and Chips)}. But this does not mean the given user profile is complete because the user may also like "Paella", although "Fish and Chips" is his/her favorite food. To simulate incomplete profiles with various degrees of incompleteness, we randomly discard attribute values from a user profile with probabilities of [0%, 10%, 30%, 50%, 70%, 90%, 100%] and obtain 7 alternative datasets, respectively.

We evaluate the performance of the full dialogue task using the following two metrics [85]:

- *response selection accuracy* (RSA): the fraction of correct responses out of all candidate responses [85, 126]; and

- *profile enrichment accuracy* (PEA): we define this metric as the fraction of correct profile values out of all discarded profile values.

We use a paired t-test to measure the statistical significance ($p < 0.01$) of relative improvements.

To compare model stability, we propose a statistic $\sigma$, namely *stability coefficient*, which is defined as the standard deviation of a list of performance results. Formally, given a list of evaluation values $[z_1, \dots, z_{N+1}]$, either RSA or PEA scores, $\sigma$ is computed as follows:

$$\begin{aligned}\sigma(\mathbf{z}) &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{z}_i - \bar{\mathbf{z}})^2} \\ \mathbf{z} &= [z_2 - z_1, \dots, z_{N+1} - z_N],\end{aligned} \tag{3.13}$$

where $\bar{z}$ is the mean of the values in performance difference list $\mathbf{z}$.

### 3.4.3 Baselines

We compare with all the methods that have reported results on the PbAbI dataset [85].

- **Memory network (MemNN).** It regards the profile information as the first user utterance ahead of each dialogue and achieves personalization by modeling dialogue context using the standard MemNN model [12].

- **Split memory network (SMemNN)**. It splits memory into a profile memory and a dialogue memory. The former encodes user profile attributes as separate entries and the latter operates the same as the MemNN. The element-wise sum of both memories are used for final decision [85].
- **Retrieval memory network (RMemNN)**. It features an encoder-encoder memory network with a retrieval module that employs the user utterances and user profiles to collect relevant information from similar users' conversations [251].
- **Personalized memory network (PMemNN)**. It uses MemNN to model the current user profile, the current dialogue history, as well as the dialogue history of all users with the same gender and age. It also models user bias towards different KB entries [126].
- **Neighbor-based personalized memory network (NPMemNN)**. Our implementation of PMemNN is based on Pytorch. Unlike PMemNN, we use the dialogue history from the nearest $(k-1)$ neighbors instead of all users with the same gender and age.

### 3.4.4 Implementation details

We follow the experimental settings detailed in [126]. The embedding size of the word/profile is 128. The size of the memory is 250. The mini-batch size is 64. The maximum number of training epochs is 250, and the number of hops is 3 (see Algorithm 1). The k-nearest neighbors (KNN) algorithm is implemented based on faiss[1] with the inner product measurement and the number of collaborative users $k = 100$. We implement NPMemNN and CoMemNN in PyTorch.[2] And the code of the other models is taken from the original papers. We use Adam [91] as our optimization algorithm with a learning rate of $0.01$ and initialize the learnable parameters with the Xavier initializer. We also apply gradient clipping [162] with range $[-10, 10]$ during training. We use $l2$ regularization to alleviate overfitting, the weight of which is set to $10^{-5}$. We treat the importance of losses of DRS and UPE equally, i.e., $\mu = 0.5$. The code is available online.[3]

## 3.5 Results (RQ2.1)

### 3.5.1 Results without discarding user profiles

We show the overall response selection performance of all methods in Table 3.2.

First, CoMemNN outperforms all baselines on both the small and large datasets by a large margin. It significantly outperforms the best baseline PMemNN by 3.06% on the small dataset and 2.80% on the large dataset. The improvements demonstrate the effectiveness of CoMemNN. We believe the main reason is that the proposed

[1] https://github.com/facebookresearch/faiss

[2] https://pytorch.org/

[3] https://github.com/Jiahuan-Pei/CoMemNN

Table 3.2: Overall performance in terms of the RSA metric. **Boldface** indicates leading results. Significant improvements over NPMemNN are marked with $^*$ (paired t-test, $p < 0.01$).

| | **Small set (%)** | **Large set (%)** |
|---|---|---|
| MemNN [85] | 77.74 | 85.10 |
| SMemNN [85] | 78.10 | 87.28 |
| RMemNN [251] | 83.94 | 87.33 |
| PMemNN [126] | 88.07 | 95.33 |
| NPMemNN | 87.91 | 97.49 |
| CoMemNN | **91.13**$^*$ | **98.13**$^*$ |

cooperative mechanism is able to enrich the incomplete profiles gradually as dialogues progress, and the enriched profiles improve help to response selection simultaneously. We will analyze this in more depth in the next session.

Second, the performance of NPMemNN is comparable to that of PMemNN on the small dataset and achieves 2.16% higher RSA on the large dataset. Recall that NPMemNN is our implementation of PMemNN using Pytorch; the only difference is the KNN algorithm used for neighbor searching, so the result shows that our new neighbor searching method is more effective. Since our CoMemNN is built upon NPMemNN, for the remaining experiments, we will use NPMemNN for further comparison and analysis.

Third, the results on the small and large datasets mostly show consistent trends. For the remaining analysis experiments in the next section (Section 3.6), we will therefore report results on the small dataset only. The findings on the large dataset are qualitatively similar.

### 3.5.2 Results with different profile discard ratios

We compare CoMemNN and NPMemNN under different profile discard ratios. The results are shown in Table 3.3.

First, CoMemNN significantly outperforms NPMemNN on both the small and large datasets when the profile discard ratios range from 0% to 90%. Specifically, it gains an improvement of 0.75%–3.79% on the small dataset and 0.64%–5.67% on the large dataset, respectively. Without discarding profile attribute values, CoMemNN achieves a 3.22% / 0.64% improvement compared with NPMemNN. Unlike the raw profiles where each attribute has only one value, the enriched profiles generated by CoMemNN are able to represent a distribution over all possible values, which can better capture users' preferences. For example, a user may label "Fish and Chips" as his favorite food, but this does not mean he does not like "Paella." With the raw profile, this is not addressed.

Second, the performance of CoMemNN steadily decreases with the increase of the profile discard ratio, as is to be expected. This is reasonable as it becomes more and more challenging for CoMemNN to find back missing values of user profiles. Interestingly, the performance difference between CoMemNN and NPMemNN first

Table 3.3: Comparison of CoMemNN and NPMemNN in terms of the RSA metric w.r.t. different profile discard ratios. **Boldface** indicates leading results. Significant improvements over NPMemNN are marked with * (paired t-test, $p < 0.01$). The values of Diff. are computed by absolute difference of RSA (%) between CoMemNN and NPMemNN.

| **Discard Ratio** | 0% | 10% | 30% | 50% | 70% | 90% | 100% |
|---|---|---|---|---|---|---|---|
| NPMemNN | 87.91 | 86.11 | 86.56 | 85.79 | 83.93 | 84.08 | **84.83** |
| CoMemNN | **91.13*** | **89.90*** | **88.69*** | **87.80*** | **86.35*** | **84.83*** | 82.85 |
| Small Set/Diff. | 3.22 | 3.79 | 2.13 | 2.01 | 2.42 | 0.75 | −1.98 |
| NPMemNN | 97.49 | 97.01 | 96.05 | 95.52 | 95.40 | 90.96 | 90.50 |
| CoMemNN | **98.13*** | **97.94*** | **97.68*** | **97.53*** | **96.98*** | **96.63*** | **92.73*** |
| Large Set/Diff. | 0.64 | 0.93 | 1.63 | 2.01 | 1.58 | 5.67 | 2.23 |

increases and then decreases with the increase of the profile discard ratio. A possible reason is that CoMemNN is able to infer the missing values of user profiles effectively with lower profile discard ratios. However, the profile enrichment ability decreases due to the absence of too many profile values. This hypothesis can be verified by the results that the increasing trend lasts longer on the large dataset. Because even with the same profile discard ratio, there are more values of user profiles left on the large dataset for CoMemNN to infer the missing ones. We note that NPMemNN outperforms CoMemNN when all user profiles are discarded on the small dataset. The reason is that UPE cannot enrich user profiles properly in this case, which results in a negative impact on DRS. But this is not the case on the large dataset where UPE can still enrich user profiles properly when the model can find enough personal information clues from more dialogue history.

Third, to answer **RQ2.4**, we compute the statistic $\sigma$ (Eq. 3.13) to compare the model stability. The $\sigma$ values for CoMemNN and NPMemNN are 0.3357/1.0407 on the small dataset and 1.3479/1.4849 on the large dataset, respectively. Thus, NPMemNN has higher deviations, which shows that CoMemNN is more stable than NPMemNN with various profile discard ratios.

## 3.6 Analysis

We analyze the performance of the following variants of CoMemNN:

- **CoMemNN**. The full model.
- **CoMemNN-PEL**. CoMemNN without profile enrichment loss (PEL), defined in Eq. 3.11.
- **CoMemNN-PEL-UPE**. CoMemNN without PEL or UPE. This is exactly NPMemNN.
- **CoMemNN-NP**. CoMemNN without the neighbor profile (NP) as input for UPE.

Table 3.4: Performance of UPE evaluated in terms of profile enrichment accuracy (PEA). In each cell, the first number represents the PEA (%), and the number in parentheses shows the difference compared with CoMemNN. ↓ and || denote a decrease and no change compared to CoMemNN, respectively.

| **Discard Ratio** | 10% | 30% | 50% | 70% | 90% | 100% |
|---|---|---|---|---|---|---|
| CoMemNN | 99.99 | 99.93 | 99.82 | 99.83 | 99.38 | 98.98 |
| CoMemNN-PEL | 85.71<br>(↓14.28) | 87.85<br>(↓12.08) | 91.34<br>(↓8.48) | 89.19<br>(↓10.64) | 90.04<br>(↓9.34) | 90.60<br>(↓8.38) |
| CoMemNN-NP | 99.87<br>(↓0.12) | 99.85<br>(↓0.08) | 99.24<br>(↓0.58) | 99.15<br>(↓0.68) | 99.13<br>(↓0.25) | 98.86<br>(↓0.12) |
| CoMemNN-NP-CP | 98.89<br>(↓1.10) | 99.09<br>(↓0.84) | 99.16<br>(↓0.66) | 99.20<br>(↓0.63) | 99.14<br>(↓0.23) | 98.92<br>(↓0.06) |
| CoMemNN-ND | 99.72<br>(↓0.26) | 99.87<br>(↓0.06) | 99.80<br>(↓0.02) | 99.46<br>(↓0.37) | 98.72<br>(↓0.66) | 97.23<br>(↓1.75) |
| CoMemNN-ND-CD | 99.99<br>(\|\|0.00) | 99.86<br>(↓0.07) | 99.68<br>(↓0.14) | 99.69<br>(↓0.14) | 99.19<br>(↓0.19) | 34.78<br>(↓64.2) |
| CoMemNN-ND-NP | 99.09<br>(↓0.90) | 98.98<br>(↓0.95) | 97.95<br>(↓1.87) | 97.69<br>(↓2.14) | 97.06<br>(↓2.32) | 97.23<br>(↓1.75) |

- **CoMemNN-NP-CP**. CoMemNN without NP or the current profile (CP) as input for UPE.
- **CoMemNN-ND**. CoMemNN without the neighbor dialogue (ND) of dialogues as input for UPE.
- **CoMemNN-ND-CD**. CoMemNN without ND or the current dialogue (CD) of dialogues as input for UPE.
- **CoMemNN-ND-NP**. CoMemNN without ND or NP of dialogues as input for UPE.

### 3.6.1 Ablation study on PEA (RQ2.2)

We study the PEA performance of different variants in Table 3.4.

First, CoMemNN can effectively enrich user profiles by inferring missing values. It is able to correctly predict more than 98.98% of missing values in user profiles under different profile discard ratios. We believe UPE benefits a lot from modeling the interaction between user profiles and dialogues. UPE is able to capture more personal information from dialogue history with dialogues gradually going on. The PEA scores are all very high, because the PbAbI dataset is simulated, which makes it relatively easy to predict missing attribute values of user profiles.

Second, we can see that each component of UPE generally has a positive effect on the performance since most PEA scores of most variants decrease. Specifically, CoMemNN-PEL decreases by 8.38%–14.20% compared with CoMemNN. This means that it is

Table 3.5: Ablation study on DRS evaluated in terms response selection accuracy (RSA). In each cell, the rst number represents the RSA (%), and the number in parentheses shows the difference compared with CoMemNN. ↓ and ↑ denote decrease and increase, respectively. Underlining marks results that are ≥1.0% higher than those of CoMemNN.

| **Discard Ratio** | 0% | 10% | 30% | 50% | 70% | 90% | 100% |
|---|---|---|---|---|---|---|---|
| CoMemNN | 91.13 | 89.90 | 88.69 | 87.80 | 86.35 | 84.83 | 82.85 |
| CoMemNN-PEL | 90.84 | 90.29 | 89.07 | 87.18 | 85.42 | 80.54 | 81.23 |
| | (↓0.29) | (↑0.39) | (↑0.38) | (↓0.62) | (↓0.93) | (↓4.29) | (↓1.62) |
| CoMemNN-PEL-UPE | 87.91 | 86.11 | 86.56 | 85.79 | 83.93 | 84.08 | 84.83 |
| | (↓3.22) | (↓3.79) | (↓2.13) | (↓2.01) | (↓2.42) | (↓0.75) | (↑1.98) |
| CoMemNN-NP | 91.06 | 91.23 | 89.17 | 85.26 | 83.30 | 82.10 | 82.83 |
| | (↓0.07) | (↑1.33) | (↑0.48) | (↓2.54) | (↓3.05) | (↓2.73) | (↓0.02) |
| CoMemNN-NP-CP | 86.60 | 86.10 | 84.56 | 83.53 | 82.48 | 81.95 | 81.35 |
| | (↓4.53) | (↓3.80) | (↓4.13) | (↓4.27) | (↓3.87) | (↓2.88) | (↓1.50) |
| CoMemNN-ND | 90.91 | 87.33 | 89.06 | 87.49 | 86.59 | 85.38 | 85.41 |
| | (↓0.22) | (↓2.57) | (↑0.37) | (↓0.31) | (↑0.24) | (↑0.55) | (↑2.56) |
| CoMemNN-ND-CD | 87.70 | 90.44 | 85.79 | 84.90 | 83.56 | 82.57 | 85.38 |
| | (↓3.43) | (↑0.54) | (↓2.90) | (↓2.90) | (↓2.79) | (↓2.26) | (↑2.53) |
| CoMemNN-ND-NP | 90.04 | 91.08 | 89.23 | 87.38 | 85.76 | 85.46 | 85.41 |
| | (↓1.09) | (↑1.18) | (↑0.54) | (↓0.42) | (↓0.59) | (↑0.63) | (↑2.56) |

important to add the UPE loss (Eq. 3.11), rather than only optimizing the DRS loss (Eq. 3.10). We also show how the four components of UPE (i.e., NP, CP, ND, and CD as defined in Section 3.3.3) affect its performance. We find that: (i) CoMemNN-ND-NP continuously decreases 0.90%–2.32% with the increase of the profile discard ratio. This means that neighbor users play an important role. (ii) CoMemNN-ND-CD (with 100% profile discard ratio) decreases dramatically, which is as expected, because CoMemNN cannot infer the missing values without any dialogue history and profiles. This also explains the increase of the corresponding RSA score in Table 3.5. (iii) The decrease is mostly less than 2.32% except that the decrease of CoMemNN-ND-CD (with 100% profile discard ratio, i.e., no NP or CP as well) is 64.2%. This reveals that different information sources are complementary to each other. The performance will not be affected largely unless all four inputs (i.e., NP, CP, ND, CD) are removed.

Lastly, we compute the stability coefficient $\sigma$ (Eq. 3.13) of the variants in Table 3.4 which are 0.1867, 1.8781, 0.2236, 0.1402, 25.6845, 0.1867, 0.4182, respectively. This shows that all variants are robust in terms of the performance of UPE with a small stability coefficient, except for CoMemNN-ND-CD.

### 3.6.2 Ablation study on RSA (RQ2.2)

We investigate the RSA performance of different variants in Table 3.5.

First, the performance decreases generally by removing any component of UPE. In particular, CoMemNN-PEL has a greater effect on RSA when the profile discard ratios

Table 3.6: Analysis of the effect of hop number on DRS. **Boldface** indicates leading results. Significant improvements over NPMemNN are marked with * (paired t-test, $p < 0.01$). The values of Diff. are computed by absolute difference of RSA (%) between CoMemNN and NPMemNN.

| **#Hop** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|
| NPMemNN | 88.11 | 87.22 | 87.91 | 87.61 |
| CoMemNN | **90.07*** | **90.78*** | **91.13*** | **90.77*** |
| Diff. | 1.96 | 3.56 | 3.22 | 3.16 |

get larger. This is reasonable because the larger the profile discard ratio, the more space for improvement the proposed model has compared with NPMemNN. CoMemNN-PEL-UPE is inferior to CoMemNN-PEL generally, which means that the UPE module helps as it implicitly impacts the DRS loss (Eq. 3.10). But this ability weakens when the profile discard ratio is larger than 90%.

Second, we observe that the four information sources (i.e., NP, CP, ND, CD) have different effects under different profile discard ratios. Particularly, the profiles of the current users and their neighbors generally contribute most to the RSA performance. We can see that CoMemNN-NP-CP drops 1.50%–4.53% under all profile discard ratios. The reason is that user profiles directly store personal information; it is easier to infer missing values from collaborative user profiles than from dialogues.

Third, we find that NP and ND are complementary to each other. CoMemNN-NP either has a massive drop (2.54%–3.05%) or small changes ($\leq$0.48%) with the most profile discard ratios, except for one obvious rise (1.33%) under the 10% profile discard ratio. In contrast, CoMemNN-ND works fine under the 10% profile discard ratio, but it performs poorly for the rest. Thus, the performance of CoMemNN is influence strongly by a drop in attribute values unless we remove both NP and ND under 100% profile discard ratios.

Lastly, the dialogue history also contributes to the RSA performance in most cases. CoMemNN-ND-CD shows decrease (2.26%–3.43%) or a small change (0.54%) for most of cases, except for an obvious increase under the 100% profile discard ratio. We think that the reason is that some of the predicted profiles are not even in the provided profiles, which leads to a very limited PEA score of 34.78% under the 100% profile discard ratio (see Table 3.4). But these predicted values happen to be useful for selecting an appropriate response in DRS.

### 3.6.3 Effect of multiple-hop mechanism (RQ2.2)

We compare the RSA performance of CoMemNN and NPMemNN with different numbers of hops. The results are shown in Table 3.6.

We see that CoMemNN greatly outperforms NPMemNN by a large margin (1.96%–3.56%) with all number of hops. This further confirms the non-trivial improvement of CoMemNN. Besides, CoMemNN improves by 1.06% when the number of hops changes from 1 to 3 and slightly decreases with 4. This means that CoMemNN benefits

Table 3.7: Analysis of profile attribute importance to DRS. **Discardd attribute** shows we discard all values of a specific attribute or a combination of two specific attributes. **Retained attribute** shows we retain all values of a specific attribute and discard all values for the rest. Italics indicate increased results compared with the result that retains all attributes. Underline indicates the lower bound baseline that retains no attributes. Boldface indicates the upper bound baseline that retains all attributes.

| **Discarded attribute** | none | gender | age | dietary | favorite | all |
|---|---|---|---|---|---|---|
| gender | / | *93.05* | *91.94* | 88.86 | *91.95* | / |
| age | / | / | *92.26* | 89.37 | 91.04 | / |
| dietary | / | / | / | 86.74 | 86.42 | / |
| favorite | / | / | / | / | 90.25 | / |
| **Retained attribute** | 82.85 | 87.46 | 87.93 | 90.57 | 87.37 | **91.13** |

Table 3.8: Analysis of profile attribute importance to DRS without the effect of neighbors. **Boldface** indicates the baseline of CoMemNN without neighbors. In each cell, the first number represents the RSA (%), the number in parenthesis shows the difference values, and ↓ denotes a decrease compared with the baseline.

| | RSA (Diff.) |
|---|---|
| CoMemNN w/o neighbors | **90.34** |
| CoMemNN w/o neighbors - gender | 88.25 (↓2.09) |
| CoMemNN w/o neighbors - age | 85.62 (↓4.72) |
| CoMemNN w/o neighbors - gender - age | 83.73 (↓6.61) |

from a multiple-hop mechanism.

### 3.6.4 Effect of different profile attributes (RQ2.3)

We explore how the four types of profile attributes (i.e., gender, age, dietary preference, and favorite food) affect the RSA performance. The results are shown in Table 3.7.

First, each attribute works well in isolation. Specifically, when we only retain the values of every single attribute, we obtain the results in the last row as 87.46%, 87.93%, 90.57%, and 87.37% for gender, age, dietary, and favorite, respectively. The attribute "dietary" contributes most followed by "age", "gender" and "favorite."

Second, different types of attributes depend on each other and influence the RSA performance differently. If we only remove the values of one attribute, we get the results on the diagonal: 93.05%, 92.26%, 86.74%, and 90.25%, respectively. Removing "dietary" drops most followed by "favorite." Thus, "dietary" contributes more than the rest.

An exception is that the RSA performance increases when discarding "gender" and "age." We believe this is the effect of the neighbors. To show this, we further investigate the effect of "gender" and "age" without using neighbor information. The results are shown in Table 3.8.

We can see that removing “gender” and “age” decreases the performance in this case. Thus, the different effects of “gender” and “age” are due to the neighbors.

## 3.7 Conclusion

In this chapter, we have studied personalized TDSs without assuming we have complete user profiles. We have proposed a cooperative memory network (CoMemNN), which introduces a cooperative mechanism to gradually enrich user profiles as dialogues progress, and to simultaneously improve response selection based on enriched profiles. We have also devised a learning algorithm to effectively learn CoMemNN with multiple hops.

Extensive experiments on the personalized bAbI dialogue (PbAbI) dataset demonstrate that CoMemNN significantly outperforms state-of-the-art baselines. Further analysis experiments confirm the effectiveness of CoMemNN by analyzing the performance and contribution of each component. Together these findings provide a positive answer to the leading research question for this chapter: multiple users can indeed collaborate successfully to improve the quality of a dialogue for each single user.

A limitation of the work presented in this chapter is that we tested the performance of CoMemNN on the only open available personalized TDSs dataset PbAbI. We encourage the community to work on creating additional resources for this task.

As to future work, we hope to experiment on more datasets and investigate how the performance varies on different datasets and whether we can further improve the performance by leveraging non-personalized TDS datasets.

Next, we turn from user collaboration to language collaboration.

# 4 Language Collaboration: Collaborative Agents for Multilingual Dialogues

In this chapter, we aim to answer the following research question:

**RQ3** Can multiple languages be used in a collaborative way to improve the performance of each single language?

We explore the idea of collaborative agents from the view of language collaboration.

We first propose a *mixture-of-languages routing* (MOLR) paradigm in a collaborative chair-experts framework: Each expert agent can be either monolingual or cross-lingual, and a chair agent conducts a mixture-of-experts for globally optimizing multilingual task-oriented dialogue systems (TDSs). Specifically, the paradigm includes four functional components, i.e., input embeddings, language model, pairwise alignment, and mixture-of-languages. We then quantify language characteristics of unity and diversity using a number of similarity metrics, i.e., genetic similarity, and word and sentence similarity based on embeddings. Our main finding is that the performance of multilingual TDSs can be greatly impacted by three key aspects, i.e., data sufficiency, language characteristics, and model design in a MOLR paradigm.

## 4.1 Introduction

How many human languages are there in the world? As of 2019, Ethnologue summarized the most extensive catalog of human languages in the world.[1] It covers 6,909 distinct languages, out of which 230 are spoken in Europe, while 2,197 are spoken in Asia.[2] Roughly 80% of the world population does not speak English [31]. English is usually regarded as a high-resource, pivot language, and English dialogue models, as well as their cross-lingual adaptations, have achieved very high performance [20, 151]. In principle, multilingual dialogue models can play a role in supporting multiple communities with multiple languages as input and output, and are highly non-trivial [53, 181, 182].

This chapter is under review as J. Pei, G. Yan, P. Ren, and M. de Rijke. Mixture-of-languages routing for multilingual task-oriented dialogue systems. *Under review*, 2022 [171].

[1] https://www.ethnologue.com/

[2] https://www.linguisticsociety.org/content/how-many-languages-are-there-world

Table 4.1: Hierarchical classification based on the Ethnologue catalog[3] for English, German, Italian, Spanish, and Thai. The Code column shows the unique identification by ISO 639-3 standards. Classification is the path to a language in the language family trees in Ethnologue.

| **Language** | **Code** | **Classification** |
|---|---|---|
| **English** | eng | Indo-European>Germanic>West>English |
| **German** | deu | Indo-European>Germanic>West>High German>German>Middle German>East Middle German |
| **Italian** | ita | Indo-European>Italic>Romance>Italo-Western>Italo-Dalmatian |
| **Spanish** | spa | Indo-European>Italic>Romance>Italo-Western>Western>Gallo-Iberian>Ibero-Romance>West Iberian>Castilian |
| **Thai** | tha | Kra-Dai>Kam-Tai>Tai>Southwestern |

First, multilingual dialogue datasets are quite scarce and face an acquisition challenge. For example, a survey [193] from several years ago reports 63 available dialogue corpora and only 2 of them contain multilingual dialogues (i.e., Verbmobil [15] and DSTC5 [90]), until March 2017. Since then, several publications have released dialogue datasets for training multilingual chitchat [32, 112], and both bilingual [113] and multilingual [41, 78, 105, 148, 190, 203, 213, 214] TDSs. Furthermore, a lack of language experts makes the acquisition of non-English data challenging [53]. For example, in the multilingual natural language understanding (NLU) dataset [190], only 11.7% and 20.0% of the utterances are obtained for Thai and Spanish, respectively, due to a lack of bilingual speakers.

Second, language commonalities and peculiarities are very important. On the one hand, languages have genetic relationships through language evolution. We list the Ethnologue catalog entries of 5 languages (i.e., English, German, Italian, Spanish, and Thai) in Table 4.1. English is neither always the best nor the only pivot language to bridge the language gap [34, 165]. On the other hand, the unity and diversity of languages are widely encoded into high-dimensional vectors in recent computational linguistics. As shown in Figure 4.2, we visualize the mT5 [242] embeddings of words from two benchmark multilingual dialogue datasets [148, 190], covering the five languages mentioned above. Thai words are clustering independently, while words from European languages are mixed up. We further conduct pairwise comparisons of the European languages in Figure 4.2. We find that intersecting areas (representing commonalities between languages) and disjoint areas (representing peculiarities of languages) can be preserved at the same time, but their proportions can be very different for different language pairs. For example, English and Spanish are not as clearly separated as the other three language pairs, so the proportion of intersecting areas is larger (see Figure 4.2).

Last but not least, the majority of TDS models focuses on either multiple language-specific optima [146, 190] or cross-lingual adaptation from English to non-English towards multilingual TDSs (see Table 4.9 and 4.10). Very few publications consider

[3] https://www.ethnologue.com/browse/names

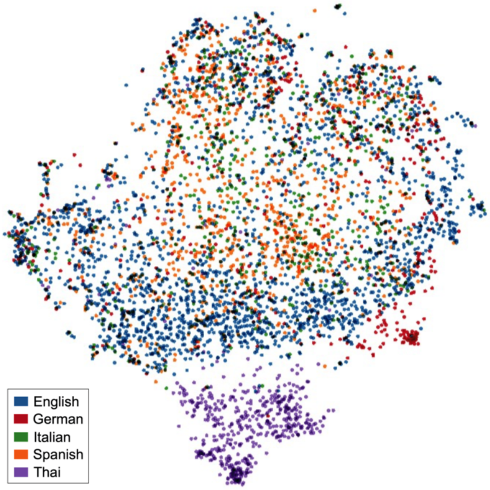


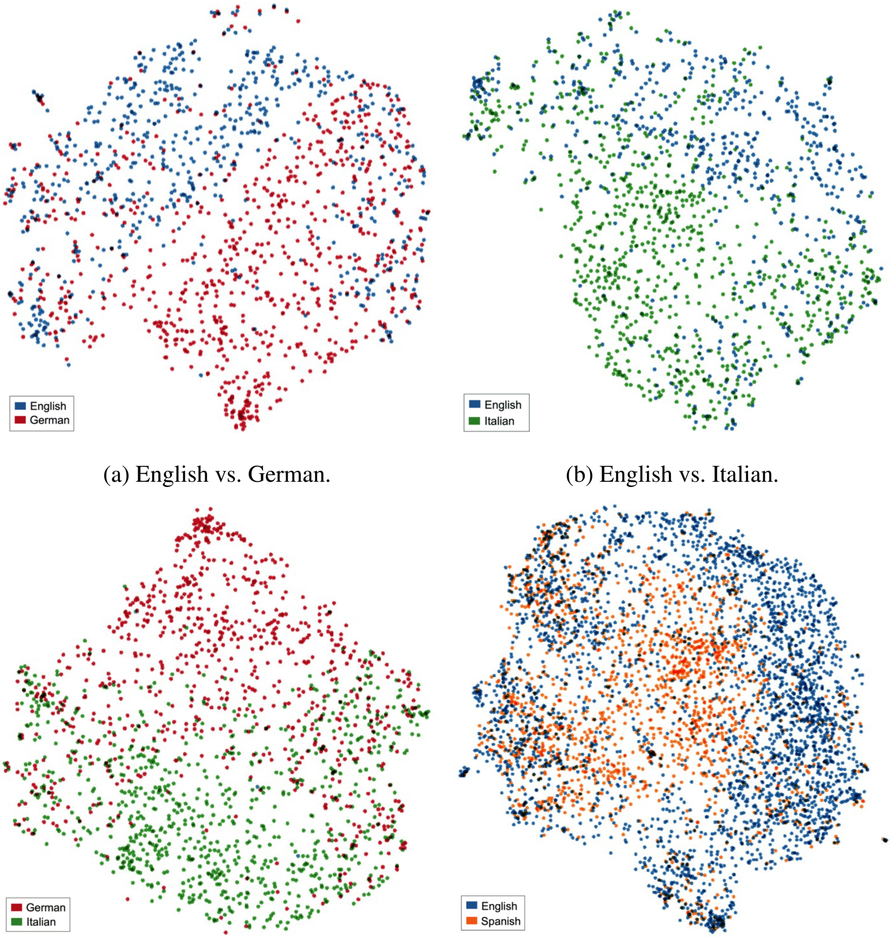
Figure 4.1: Visualization of the embeddings of words from two benchmark multilingual dialogue datasets, covering English, German, Italian, Spanish, and Thai. We conduct dimension reduction using the UMAP algorithm [140] and plot all scatter in 2D coordinates using the Tensorflow embedding projector.[4]

improving multilingual performance simultaneously, but simply training models using multilingual data does not always lead to improvements, e.g., multilingual NBT [148] and bilingual mBART [113] are outperformed by their monolingual settings in terms of multiple evaluation metrics. Besides, optimizing all pipeline tasks requires all language-specific annotations, which makes global optimization more challenging [181].

In this chapter, we propose a multilingual dialogue paradigm, as shown in Figure 4.3, which aims to: (i) fully makes use of multilingual data; (ii) captures commonalities between, and peculiarities of, languages, and (iii) improves multilingual performance simultaneously.

We recast the multilingual TDS problem in a collaborative TDS framework [166, 168]: $k$ expert agents account for monolingual and cross-lingual dialogues, and a chair agent conducts a mixture-of-experts for globally optimizing multilingual dialogues. To be more precise, we unify TDS tasks as a standard dialogue generation task and implement a mixture-of-languages routing (MOLR) paradigm with four functional components, i.e., (i) input embeddings, (ii) language model, (iii) pairwise alignment, and mixture-of-languages. For the former two components, we choose mT5 [242] as the backbone of our base model after comparing with pre-trained language baselines [19, 261]. Note that each base model can be either a monolingual or cross-lingual expert agent, and it can flexibly be replaced by other popular multilingual language models such as mBERT [39], mBART [118], and XLM-R [28], etc. Next, we introduce pairwise alignment to bridge the relationship between every two language routes. Here, a *language route* is a path commencing from a source language as the starting point,

[4] https://projector.tensorflow.org/



(a) English vs. German.

(b) English vs. Italian.

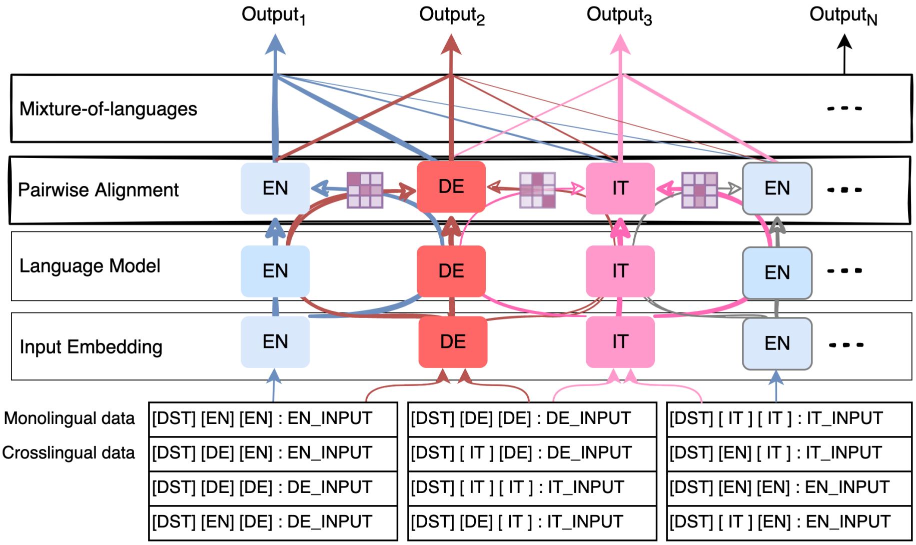
(c) German vs. Italian.

(d) English vs. Spanish.

Figure 4.2: Pairwise comparison of the embeddings of words in dialogues from European languages.

passing through a pivot language, to a target language as its destination. Language commonalities and peculiarities can be embedded into pairwise alignment states. After that, we conduct global optimization by the mixture-of-languages routing with two collaboration policies, i.e., route-addressing and parameter-sharing. By *mixture of languages routing* we mean the process of learning a combination of routes in the proposed model between or across multiple languages. This setup enables the multilingual dialogue model to automatically learn the pivot languages, rather than fixing English as the only pivot language. Moreover, the unified generation framework equips the proposed model with the ability to optimize multiple subtasks, simultaneously.

To verify the effectiveness of the proposed MOLR paradigm, we conduct extensive



| Monolingual data | [DST] [EN] [EN] : EN_INPUT | [DST] [DE] [DE] : DE_INPUT | [DST] [ IT ] [ IT ] : IT_INPUT |
|---|---|---|---|
| Crosslingual data | [DST] [DE] [EN] : EN_INPUT | [DST] [ IT ] [DE] : DE_INPUT | [DST] [EN] [ IT ] : IT_INPUT |
| | [DST] [DE] [DE] : DE_INPUT | [DST] [ IT ] [ IT ] : IT_INPUT | [DST] [EN] [EN] : EN_INPUT |
| | [DST] [EN] [DE] : DE_INPUT | [DST] [DE] [ IT ] : IT_INPUT | [DST] [ IT ] [EN] : EN_INPUT |

Figure 4.3: The paradigm of mixture-of-languages routing (MOLR) in multilingual TDSs. Taking the dialogue state tracking (DST) task as an example, the raw inputs are extended with prefixes and processed into monolingual and cross-lingual data, respectively.

experiments on two benchmark datasets, i.e., the multilingual DST dataset [148] and the NLU dataset [190]. We find that our models are on par with and even outperform state-of-the-art baselines for both multilingual DST and NLU tasks. At best, compared with mT5, the proposed MOLR models improve 2.31%/2.56%/0.67% of joint goal accuracy for English/German/Italian on the DST task, and 0.13%/1.89%/5.53% of slot F1 for English/Spanish/Thai on the NLU task. Note that most of the baselines conduct classification over the predefined task-related label space; in contrast, we generate all the labels from the vocabulary space.

The larger prediction space increases the difficulty of tasks, but the benefits are obvious: our paradigm is able to predict values that are not predefined and is applicable to all dialogue tasks in a unified way.

The main contributions of this chapter are as follows:

- We propose a mixture-of-languages routing (MOLR) paradigm, which is able to globally and simultaneously optimize the multilingual task-oriented dialogue system (TDS) performance. MOLR benefits from multilingual data argumentation, language characteristic modeling, and mixture-of-language routing.

- We develop generation baselines that are at least on par with the state-of-the-art classification baselines.

- We carry out a large number of contrastive experiments and deep-dive analyses, which reveal the effectiveness of the MOLR paradigm and help understand its effectiveness.

- We find that it is better to gradually cross the language chasm: a larger degree of similarity between the source language and pivot language is usually helpful for the overall performance.

## 4.2 Related work

Given the challenges of multilingual TDSs, we summarize related work from three points of view: (i) data, (ii) language, and (iii) model.

### 4.2.1 Multilingual data augmentation

Data augmentation has been widely used for alleviating data scarcity problems in multilingual dialogues [183]. On the one hand, data augmentation targets better representation of dialogues. Zhao et al. [257] use atomic templates to produce exemplars from dialogue acts, followed by a sentence generator to complete the whole utterance. Louvan and Magnini [123] involve simple text and syntax substitutions, and combine them with pre-trained language models. Yin et al. [248] replace text spans with paraphrases and use reinforcement learning to control the quality of the augmented data. Pei et al. [169] search nearest neighbor dialogues as supplements to current dialogue to alleviate the scarcity of user preferences. Yan et al. [243] introduce heuristic approaches to generate data and adopt contrastive learning to further improve the overall performance. Most recent work usually benefits from monolingual pre-trained language models (e.g., BERT [39] and GPT-2 [176]).

On the other hand, data augmentation aims to bridge language gaps. Dominant code-switching methods [96, 174] translate sentences in English into randomly selected target languages, which enables them to fine-tune multilingual transformers with generalization ability across languages. XeroAlign [63] introduces an auxiliary loss function based on machine translation and jointly optimizes the overall performance with the primary task. Kaliamoorthi et al. [88] conduct knowledge transfer during distillation from a pre-trained mBERT teacher to a tiny student model. Mrkšić et al. [148] learn specialized cross-lingual vector spaces by multilingual data training enhanced with semantic relations from lexical resources. Most recent work crosses the language chasm using multilingual pre-trained language models (e.g., mBERT [39, 41, 78], XLM-R [28, 78, 105, 261], mT5 [214, 264]).

Similar to most recent work [214, 264], we choose a state-of-the-art multilingual pre-trained language model (i.e., mT5) as our backbone for both better dialogue representation and language transfer. But unlike the above approaches, (i) we generate pairwise language routes and focus on how to learn the relationships between language pairs, and (ii) we aggregate language routes for global optimization of multilingual TDSs.

### 4.2.2 Unity and diversity of languages

In bioscience [50] and linguistic studies [42, 52, 210, 215], both unity and diversity play key roles for cross-linguistic variation in human languages.

Over time, languages generate biological or genetic relationships [67]. Linguists and language institutions have conducted large-scale studies on language affinity[5] and the Ethnologue catalog.[6] Generally, a language family tree is a common way to interpret genetic relationships that can reveal the unity and diversity of languages [208]. Their basic assumption is that two languages belong to the same language family if they are from a common ancestor, or one is descended from the other.

In modern linguistics, important research topics include universal grammar [35, 152] and linguistic typology [159, 215, 230]. The former focuses on unity, in which all languages are treated as universal components of the language faculty [35]. This is the theoretical basis of research on part-of-speech tagging [16, 150], chunking [84, 131], and syntactic parsing [128, 141]. The latter emphasizes diversity, which captures the structural differences of languages, as the principal bridge, to discover universals [36]. Morphology is usually diverse across languages, and it is hard to find universals for traditional linguistic typology [189]. The world language tree is constructed based on Levenshtein distances, which define the average number of edits needed to convert a source language to a target language [149].

Language similarity has been a commonly used metric to quantitatively measure unity and diversity in recent computational linguistics [8, 18, 225]. One branch of work measures language similarity by their structural properties [33]. Bjerva et al. [10] define language similarity based on language structures, i.e., phrase structure trees and dependency relations. Oco et al. [156] compute Dice's coefficient to measure the similarity of eight Philippine languages based on the language family tree in the Ethnologue. However, these approaches do not apply when the structure is not available. Another branch of work measures language similarity as the lexical overlap between languages based on handcrafted cognates [154] or automatically extracted cognates [194]. Beinborn et al. [9] identify cognates based on character-based machine translation. However, their methods cannot compare the similarity of cognates without a translation relationship (e.g., English "father" and the Italian "padre") [8]. To this end, most recent work encodes natural languages into high dimensional vectors namely embeddings, e.g., word embeddings [187, 199] and word-based syntax embeddings [109] and pretrained language models [95, 161]. Therefore, the unity and diversity of languages can be measured using the similarity and dissimilarity of embeddings.

In this chapter, we conduct an analysis of multilingual TDS results from the point of view of language characteristics, i.e., the unity and diversity of languages. We compare commonalities and specifications of languages using multiple aspects, including visualization of word embeddings, as well as genetic and embedding-based similarity metrics.

### 4.2.3 Multilingual TDS models

Monolingual TDSs have achieved considerable progress as reported in a large number of recent publications [20, 151, 193, 255]. Many recent studies have built new datasets and/or tasks to advance research on multilingual TDSs [41, 78, 160, 241]. However, it is hard to fairly compare with the majority of approaches because they do not report

[5] `http://www.linguaechristi.org/people-groups/`

[6] `https://www.ethnologue.com/browse/names`

results on those all datasets [181]. Thus, in this work, we mainly focus on a comparison of cross-lingual and multilingual models on two commonly-used tasks (i.e., DST [148] and NLU [190]).

### Cross-lingual models

Existing cross-lingual models mainly consider two key factors: dialogue representation and cross-lingual transfer. To conduct better language modeling, previous studies utilize variants of sequential models. Upadhyay et al. [213] jointly train bilingual embeddings with a biRNN model for few-shot cross-lingual NLU. Liu et al. [119] equip biLSTM model with latent variables and word pairs to refine the aligned cross-lingual word embeddings. Schuster et al. [190] deploy a biLSTM-CRF model, where the cross-lingual transfer comes from sharing between the biLSTM and CRF layer across languages. Liu et al. [121] develop a biLSTM, transformer, and mBERT for sequence labeling tasks and find that removing the word order can improve cross-lingual performance. Several researchers generate code-switching sentences to enable cross-lingual capabilities, by either replacing words [83, 120] or sentences [174] in target languages. MultiATIS++ [241] learns slot alignment based on an mBERT encoder, machine translation, and label projection. GlobalWoZ [41] introduces several data augmentation baselines for zero-shot and few-shot cross-lingual learning on the proposed dataset. Siddhant et al. [198] gain cross-lingual transfer capabilities by representations from a multilingual neural machine translation encoder. Gritta and Iacobacci [63] use an auxiliary translation-based loss function to jointly learn with the primary task. Xiang et al. [238] inject multi-granularity translation-based noise to improve the robustness of cross-lingual task-oriented dialogues. Hung et al. [78] finetune the XLM-R model with English and target languages in zero-shot and few-shot transfer settings. Li et al. [105] provide several multilingual pretrained benchmarks such as XLM-R and mBAR, and evaluate them on the multilingual ATIS and MTOP datasets for task-oriented semantic parsing. Very recently, Zuo et al. [264] apply mT5 with meta-learning on their unpublished dataset, and Van et al. [214] simply use mT5 as a state-of-the-art benchmark in their Vietnamese task-oriented dialogue dataset.

To sum up, the proposed methods enable transfer across languages using a variety of techniques, including cross-lingual word embeddings [213], multilingual knowledge distillation [23], transferable latent variables [119], code-switching [83, 120, 174], word alignment [120, 241], and machine translation [63, 198, 238]. Most recent work benefits from these techniques and from pre-trained multilingual language models such as mBERT [39, 41], XLM-R [78, 105, 261], andmT5 [214, 264].

### Multilingual models

Only a few previous studies target multilingual TDS models. An intuitive solution is to train a single model on combined multilingual datasets and evaluate the model on test data for all languages, respectively. Mrkšić et al. [148] use constraints from monolingual and cross-lingual synonymy and antonymy to finetune multilingual word embedding spaces and apply them to the DST task. Schuster et al. [190] use a multilingual translation-based biLSTM encoder to learn contextual word representations, evaluating multiple languages. GlobalWoZ [41] introduces several data augmentation baselines for zero-shot and few-shot cross-lingual learning on the proposed dataset. Ding et al. [41] monolingual and cross-lingual use cases are parts of multilingual TDSs and optimize for each use case separately. Recent work by Zuo et al. [264] reports a benchmark of mT5 with meta-learning [49]; however, neither the dataset nor the source code of the model is publicly accessible.

None of the released models has modeled language relationships or conducted global optimization for multilingual TDSs on public datasets, to the best of our knowledge. Unlike the majority of classification models, the proposed generation model (i.e., MOLR) achieves competitive performance and is able to predict out-of-ontology slot values as in [97, 235, 240].

## 4.3 Collaborative multilingual dialogue paradigm

### 4.3.1 A unified task-oriented dialogue system

A dialogue consists of multiple turns between a user and a system. At the $t$-th turn, the user provides an utterance $U_t$, and the system produces a response $R_t$ as a reply. To get high-quality responses $R_t$, a TDS is usually decomposed into four subtasks: natural language understanding (NLU), dialogue state tracking (DST), dialogue policy learning (DPL) and natural language generation (NLG).

In this work, we unify a TDS with a neural network $f_\theta(.)$ parameterized by $\theta$, which generally contains (i) an input embedding, (ii) hidden states encoding, and (iii) output projection layers. This neural network works with all subtasks in an end-to-end fashion. Specifically, we formulate the four subtasks as follows.

#### Natural language understanding (NLU)

Given a current user utterance $U_t$ as input, the model outputs intents $I_t$ and slots $S_t$ by:

$$I_t, S_t = f_\theta(U_t). \tag{4.1}$$

#### Dialogue state tracking (DST)

Given a dialogue history $C_t = [U_1, S_1, \ldots, U_t]$ as input, the model outputs a belief state $B_t$ by:

$$B_t = f_\theta(C_t), \tag{4.2}$$

which can be denoted as a set of triples representing slot-value pairs for a specific domain: (domain, slot_name, value).

#### Dialogue policy learning (DPL)

Given dialogue history $C_t$, belief states $B_t$, and retrieval records from database $D_t$ as input, the DPL outputs system actions by:

$$A_t = f_\theta([C_t; B_t; D_t]), \tag{4.3}$$

which is a list of triples representing as (domain, action_type, slot_name).

### Natural language generation (NLG)

Given dialogue history $H_t$, belief states $B_t$, retrieval records from database $D_t$, and system actions $A_t$ as input, the model outputs a response $R_t$ by:

$$R_t = f_\theta([C_t; B_t; D_t; A_t]). \tag{4.4}$$

To unify the above subtasks, we tackle it as a sequence-to-sequence generation task [75]. The input of all tasks is a sequence of tokens that are aggregated from the concatenation of input sources, i.e., $[U_t]$, $[C_t]$, $[H_t; B_t; D_t]$ $[H_t; B_t; D_t; A_t]$ for NLU, DST, DPL, NLG, respectively.

## 4.3.2 Monolingual and cross-lingual expert agents

We use mT5 [242] as our backbone following conditional causal language modeling [243], which adopts a transformer-based encoder-decoder model to learn a mapping $f$ from an input sequence $\mathbf{X}_{1:n} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ to a target sequence $\mathbf{Y}_{1:m} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m)$, i.e., $f_{\theta_{\text{enc}}, \theta_{\text{dec}}} : \mathbf{X}_{1:n} \rightarrow \mathbf{Y}_{1:m}$, by the following conditional probability distribution:

$$p_{\theta_{\text{enc}}, \theta_{\text{dec}}}(\mathbf{Y}_{1:m} | \mathbf{X}_{1:n}). \tag{4.5}$$

For each input sequence, the encoder converts $X_{1:n}$ to the corresponding hidden states $\tilde{\mathbf{X}}_{1:n} = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_n)$, the encoder is represented as $f_{\theta_{\text{enc}}} : \mathbf{X}_{1:n} \rightarrow \tilde{\mathbf{X}}_{1:n}$, formally, the probability can be computed as:

$$p_{\theta_{\text{enc}}}(\tilde{\mathbf{X}}_{1:n} | \mathbf{X}_{1:n}). \tag{4.6}$$

Mathematically, the decoder learns the probability distribution of $Y_{1:m}$ given $H_{1:n}$, i.e., $p_\theta(\mathbf{Y}_{1:m} | \tilde{\mathbf{X}}_{1:n})$. Using Bayes's rule, the distribution can be decomposed into a conditional distribution over the vocabulary $\mathcal{V}$ at the $j$-th timestamp token in the target sequence by:

$$p_{\theta_{dec}}(\mathbf{Y}_{1:m} | \tilde{\mathbf{X}}_{1:n}) = \prod_{j=1}^{m} p_{\theta_{dec}}(\mathbf{y}_j | \mathbf{Y}_{0:j-1}, \tilde{\mathbf{X}}_{1:n}), \tag{4.7}$$

where $\mathbf{y}_0$ denotes the 0-th target vector that represents the vector of the special "begin-of-sentence" token [BOS]. The model can be learned by minimizing the cross-entropy loss as follows:

$$\mathcal{L}_{expert} = -\sum_{i=1}^{N} \sum_{j=1}^{n_i} \mathbf{y}_j^i \log p_\theta(\mathbf{y}_j^i | \mathbf{Y}_{0:j-1}^i, \tilde{\mathbf{X}}_{1:n}^i), \tag{4.8}$$

where $N$ denotes the batch size and $n_i$ denotes the length of the $i$-th target sequence.

For a monolingual agent, both the input sequence $\mathbf{X}_{1:n}$ and the target sequence $\mathbf{Y}_{1:m}$ are in the same language. For a cross-lingual agent, the input sequence $\mathbf{X}_{1:n}$ and the target sequence $\mathbf{Y}_{1:m}$ are from two different languages.

### 4.3.3 Multilingual agents with mixture-of-languages routing

We introduce the workflow of the MOLR model as shown in Figure 4.3, considering the DST task as an example. First, we follow T5's modeling of prefix and use "[TASK]" as the class label [177] and extend each raw input with a task-specific prefix in the following format:

[TASK] [Pivot-language] [Target-language]: [Source-language-input]

Note that if [Pivot-language] and [Target-language] are identical, then the processed data is monolingual data, otherwise it is cross-lingual data. Then, the processed inputs pass through the input embedding layers followed by a language model, and they are transformed into language-specific hidden states. Next, MOLR uses a learnable matrix to conduct pairwise alignment for every two language-specific hidden states. Last, MOLR adopts mixture-of-languages policies to integrate all states from multiple routes between or across multiple languages. To be more specific, we implement the input embeddings layers and the language model based on mT5, the state-of-the-art pretrained language model, and introduce pairwise alignment and mixture-of-languages routing as follows.

#### Pairwise alignment

Recall that in Eq. 4.7, the $k$-th monolingual model outputs the probability over the vocabulary $\mathcal{V}$ at the $j$-th timestamp by:

$$\begin{aligned}
&p^{k}_{\theta_{dec}}(\mathbf{y}_j|\mathbf{Y}_{0:j-1},\tilde{\mathbf{X}}_{1:n})\\
=&\operatorname{softmax}(f_{\theta_{\text{dec}}}(\mathbf{Y}_{0:j-1},\tilde{\mathbf{X}}_{1:n}))\\
=&\operatorname{softmax}(\psi_{\theta_{\text{task}}}(\tilde{\mathbf{y}}^{k}_{j}))\\
=&\operatorname{softmax}(\mathbf{W}^{\top}_{\text{emb}}\tilde{\mathbf{y}}^{k}_{j})\\
=&\operatorname{softmax}([\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_{|\mathcal{V}|}]^{\top}\tilde{\mathbf{y}}^{k}_{j}),
\end{aligned} \tag{4.9}$$

where $\tilde{\mathbf{y}}^{k}_{j} \in \mathbb{R}^{d}$ represents the decoded hidden state at the $j$-th timestamp from a language model. Hereby we use a pre-trained language model, mT5. $\psi_{\theta_{\text{task}}}$ denotes the task layer and $\mathbf{W}_{\text{emb}} = [\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_{|\mathcal{V}|}] \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the word embedding matrix.

Given any two monolingual models for languages $l_a, l_b$, the hidden states can be denoted as $\tilde{\mathbf{y}}^{\prime a}_{j}$ and $\tilde{\mathbf{y}}^{\prime b}_{j}$. Given a learnable matrix $\mathbf{M}^{a\to b}_{j} \in \mathbb{R}^{d\times d}$ that transforms the decoded hidden states from $\mathbf{y}^{a}_{j}$ to $\tilde{\mathbf{y}}^{\prime b}_{j}$, and vice versa, formally, we can denote the pairwise alignment as:

$$\begin{aligned}
\tilde{\mathbf{y}}^{\prime b}_{j} &= \mathbf{M}^{a\to b}_{j}\tilde{\mathbf{y}}^{a}_{j} \in \mathbb{R}^{d},\\
\tilde{\mathbf{y}}^{\prime a}_{j} &= \mathbf{M}^{b\to a}_{j}\tilde{\mathbf{y}}^{b}_{j} \in \mathbb{R}^{d}.
\end{aligned} \tag{4.10}$$

The benefit of this transition is that we can learn the hidden state of language $b$ even though we only have the training data of language $a$ and vice versa.

### Mixture-of-languages routing

To learn from a mixture of language routes, we utilize two collaboration policies, i.e., route-addressing and parameter-sharing.

**Route-addressing.** Let $H = [\tilde{\mathbf{y}}_j^a; \tilde{\mathbf{y}}_j'^a; \tilde{\mathbf{y}}_j^b; \tilde{\mathbf{y}}_j'^a; \ldots] \in \mathbb{R}^{l \times d}$ ($\frac{l}{2}$ is the number of languages and $d$ is the dimension), and $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times d}$ be the matrices for query $Q$, key $K$, and value $V$. Each $H$ is associated with a query $Q$ and a key-value pair $(K, V)$. The computation of an attentive representation $A$ of $\mathbf{y}_j$ in the self-attention is:

$$
\begin{aligned}
&Q = \mathbf{W}_q H \in \mathbb{R}^{l \times d}, K = \mathbf{W}_k H \in \mathbb{R}^{l \times d}, V = \mathbf{W}_v H \in \mathbb{R}^{l \times d},\\
&A = \mathrm{softmax}(\alpha^{-1} Q K^{\top}) \in \mathbb{R}^{l \times l},\\
&\tilde{\mathbf{y}}_j = \phi(AV) \in \mathbb{R}^{d},
\end{aligned}
\tag{4.11}
$$

where $H$ is the attended output and $A$ is the attention distribution that attends to $V$, $\alpha$ is a scaling factor, and $\phi$ is a linear layer followed by the accumulation of attended values, parameterized by $\theta$.

**Parameter-sharing.** For the same task and the same language, all the model parameters are shared, otherwise only the parameters $\theta_{\text{task}}$ in a task layer $\psi$ (see Eq. 4.9) are not shared, and the other parameters in the model are shared. In the shared modules, we aim to learn a common space representation for all tasks. This policy serves as regularization and alleviates the over-fitting problem, as the model learns a representation that generalizes to all tasks.

## 4.4 Experimental setup

### 4.4.1 Research questions

We seek to answer the following questions in the experiments:

(**RQ4.1**) Does the mixture-of-languages routing (MOLR) model improve the performance of monolingual and multilingual models?

(**RQ4.2**) How do language characteristics influence the performance of MOLR models? (i) How to qualitatively analyze language unity and diversity? (ii) How to quantify language unity and diversity? (iii) How do language unity and diversity influence the mixture of languages?

(**RQ4.3**) How do the key components influence the performance of MOLR models? (i) How do different combination policies influence the MOLR model? (ii) How do the different number of layers of expert agents influence the gains of the MOLR model?

### 4.4.2 Datasets and evaluation

We conduct a large number of experiments on two benchmark datasets for the following multilingual TDS tasks to fairly compare with the majority of prior approaches [181].

### Dialogue state tracking (DST)

The multilingual DST dataset [148] is extended from the WOZ 2.0 dataset [229] by manually translating English into Italian and German, respectively. For each language, the dataset contains 1200 multiple-turn dialogues in the restaurant domain, and it is split into 600, 200, and 400 dialogues for training, validation, and testing. The dataset contains 4 types of goal-related slots: 3 informing slots (i.e., food, price range, and area) to track a user's search constraints, and 1 request slot (i.e., request) to track a user's questions about the search results. The evaluation metrics are:

- *Joint goal accuracy*, which measures the proportion of dialogue turns where all search constraints exactly match the ground truth on the test set.
- *Request accuracy*, which represents the proportion of dialogue turns where all the user questions are recognized correctly.

### Natural language understanding (NLU)

The multilingual NLU dataset [190] consists of 43k, 8.6k, and 5k single-turn dialogues in English, Spanish, and Thai, respectively, covering 3 domains (weather, alarm, and reminder). The dataset has 12 types of intents and 11 types of slots. The evaluation metrics are:

- *Intent accuracy*, which indicates the proportion of the correctly identified intents.
- *Slot F1*, which is the geometric mean of the precision and recall for slot filling.

### Language similarity metrics

We propose language similarity metrics to compare the similarity of any two languages $\alpha, \beta$ from a genetic and semantic point of view. A higher degree of similarity denotes a higher degree of language unity, while a smaller degree of similarity denotes a higher degree of language diversity. To compare phylogenetic relationships, the Robinson-Foulds distance is the most widely used metric [164]. To measure semantic similarity, word and sentence embeddings are widely used in modern NLP tasks [18].

- *Genetic similarity*, which defines the similarity of any two languages based on their Robinson-Foulds distance (RFD)[7] in language family trees. Here we define it as $\phi_{genetic}(\alpha, \beta) = \frac{1}{\text{RFD}(\alpha,\beta)+1}$ if they have at least one ancestor, otherwise $\phi_{genetic}(\alpha, \beta) = 0$. $\text{RFD}(\cdot, \cdot)$ counts the number of unique entries that are not in common in the classification based on the Ethnologue catalogue (see Table 4.1).
- *Word similarity*, which measures the parallel degree of two languages using the cosine similarity of the centroid word embeddings of the datasets, i.e., $\phi_{word}(\alpha, \beta) = \cos(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$. We compute the centroid of word embeddings $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ as the mean of all word embeddings.

[7] https://en.wikipedia.org/wiki/Robinson%E2%80%93Foulds_metric

- *Sentence similarity*, which measures the parallel degree of two languages as the cosine similarity of the centroid sentence embeddings in the datasets, that is, $\phi_{sentence}(\alpha, \beta) = \cos(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$. A language can be represented by the mean of all sentence embeddings in a dataset. We compute the centroid of word embeddings $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ as the mean of all sentence embeddings. Here we use the embedding of the "[TASK]" token at the beginning of a sentence as its sentence embedding.

### 4.4.3 Language routes of mT5 and variants

Recall that a language route is a path starting from a source language to a target language, passing through a pivot language. We format all language routes of mT5 and its variants in Table 4.2. The notation $I(\cdot)$, $H(\cdot)$, $T(\cdot)$, and $M(\cdot)$ indicates input layers, hidden layers, task layers, and mapping layers (see Section 4.3.2 and 4.3.3). In this chapter, we develop $I(\cdot)$ and $H(\cdot)$ with a language model (Eq. 4.5) and $M(\cdot)$ for pairwise alignment (Eq. 4.10). To be more specific, we have the following types of language routes for mT5 and its variants:

- **mT5**: single language route for monolingual models.

- **mT5+bDA**: double language routes for training a single model with bilingual data.

- **mT5+bMOLR**: quadruple language routes for training a bilingual model with bilingual data. There are two monolingual routes and two cross-lingual routes.

- **mT5+bDA**: multiple language routes for training a single model with multilingual data. Here we use triple language routes.

- **mT5+bDA**: multi-hop quadruple language routes for training a multilingual model with multilingual data in multiple stages. Here we use two stages of quadruple language routes. The model from Hop1 is used as a pre-trained model for Hop2.

### 4.4.4 Baselines

For the DST task, we consider four groups of baselines, depending on the base model that they use: (i) based on neural belief tracker (NBT), (ii) based on global-locally self-attentive dialogue state tracker (GLAD), (iii) based on bidirectional encoder representations from transformers (BERT), and (iv) based on cross-lingual language model pretraining (XLM). The selection is based on recent DST models that regard English, German, and Italian as target languages, and report comparable results on the multilingual DST dataset [148].

For the NLU task, we also consider four groups of baselines, depending on the base model that they use: (i) based on recurrent neural networks (RNNs), (ii) based on transformers, (iii) based on bidirectional encoder representations from transformers (BERT), and (iv) based on cross-lingual language model pretraining (XLM). The

Table 4.2: Language routes of the proposed models given any three languages $\alpha$, $\beta$, $\gamma$ with DST or NLU as a [TASK].

| Model | Setting | Language routes |
|---|---|---|
| mT5 | Single language route | [TASK]$[\alpha][\alpha]: I(\alpha) \rightarrow H(\alpha) \rightarrow T(\alpha)$ |
| mT5+bDA | Double language routes | [TASK]$[\alpha][\alpha]: I(\alpha) \rightarrow H(\alpha) \rightarrow T(\alpha)$<br>[TASK]$[\beta][\beta]: I(\beta) \rightarrow H(\beta) \rightarrow T(\beta)$ |
| mT5+bMOLR | Quadruple language routes | [TASK]$[\alpha][\alpha]: I(\alpha) \rightarrow H(\alpha) \rightarrow T(\alpha)$<br>[TASK]$[\beta][\beta]: I(\beta) \rightarrow H(\beta) \rightarrow T(\beta)$<br>[TASK]$[\beta][\alpha]: I(\alpha) \rightarrow H(\beta) \rightarrow M(\beta \rightarrow \alpha) \rightarrow T(\alpha)$<br>[TASK]$[\alpha][\beta]: I(\beta) \rightarrow H(\alpha) \rightarrow M(\alpha \rightarrow \beta) \rightarrow T(\beta)$ |
| mT5+mDA | Multiple language routes | [TASK]$[\alpha][\alpha]: I(\alpha) \rightarrow H(\alpha) \rightarrow T(\alpha)$<br>[TASK]$[\beta][\beta]: I(\beta) \rightarrow H(\beta) \rightarrow T(\beta)$<br>[TASK]$[\gamma][\gamma]: I(\gamma) \rightarrow H(\gamma) \rightarrow T(\gamma)$ |
| mT5+mMOLR | Multi-hop language route | Hop 1:<br>[TASK]$[\gamma][\gamma]: I(\gamma) \rightarrow H(\gamma) \rightarrow T(\gamma)$<br>[TASK]$[\beta][\beta]: I(\beta) \rightarrow H(\beta) \rightarrow T(\beta)$<br>[TASK]$[\beta][\gamma]: I(\gamma) \rightarrow H(\beta) \rightarrow M(\beta \rightarrow \gamma) \rightarrow T(\gamma)$<br>[TASK]$[\gamma][\beta]: I(\beta) \rightarrow H(\gamma) \rightarrow M(\gamma \rightarrow \beta) \rightarrow T(\beta)$<br>Hop 2:<br>[TASK]$[\alpha][\alpha]: I(\alpha) \rightarrow H(\alpha) \rightarrow T(\alpha)$<br>[TASK]$[\beta][\beta]: I(\beta) \rightarrow H(\beta) \rightarrow T(\beta)$<br>[TASK]$[\beta][\alpha]: I(\alpha) \rightarrow H(\beta) \rightarrow M(\beta \rightarrow \alpha) \rightarrow T(\alpha)$<br>[TASK]$[\alpha][\beta]: I(\beta) \rightarrow H(\alpha) \rightarrow M(\alpha \rightarrow \beta) \rightarrow T(\beta)$ |

selection is based on recent NLU models that regard English, Spanish, and Thai as target languages, and report comparable results on the multilingual DST dataset [190].

### 4.4.5 Implementation details

We use a pre-trained model mT5-small from the Huggingface library.[8] It consists of 8 layers of transformer blocks for both encoders and decoders. Each attention module has 6 attention heads, and the scaling factor $\alpha$ is 1. The total number of parameters is about 300 million.

We set the training epochs to 60. We use the AdamW optimizer [122] with the default learning rate of 1e-5. We use a linear scheduler with 2,000 warmup steps. In the DST task, we set the batch size to 6 and gradient accumulation to 2. In the NLU task, we set the batch size to 16 and gradient accumulation to 1.

## 4.5 Results

We show experimental results to answer the research questions in Section 4.4.1.

[8] https://huggingface.co/google/mt5-small

Table 4.3: Comparison of dialogue state tracking (DST) models for supervised learning using English, German, and Italian as target languages. In the cells with results, the numbers before and after "/" denote joint the goal accuracy and request accuracy, respectively. Boldface indicates leading results.

| | | **Joint Goal/Request Accuracy (%)** | | |
|---|---|---|---|---|
| **DST Models** | **Settings** | **English** | **German** | **Italian** |
| **NBT** | | | | |
| NBT [147] | NBT w/CNN encoder | 84.20/91.60 | – | – |
| NBT-DNN [147] | NBT w/DNN encoder | 84.40/91.20 | – | – |
| NBT+SSU [146] | NBT+statistical state update | 84.80/– | 68.10/– | 76.10/– |
| StateNet [184] | NBT+LSTM-based state update | 88.90/– | – | – |
| NBT+Morph [218] | NBT+morphology fine-tuning | – | 66.30/– | 78.10/– |
| NBT+mDA [148] | NBT+multilingual data augmentation | 82.80/– | 57.70/– | 77.10/– |
| **GLAD** | | | | |
| GLAD [260] | Global-locally self-attentive DST | 88.10/97.10 | – | – |
| GCE [153] | GLAD+globally-conditioned encoder | 88.51/97.38 | – | – |
| GLAD+DA [248] | GLAD+paraphrase data augmentation | 88.00/– | – | – |
| GLAD+RDA [248] | GLAD+DA+reinforcement learning | 90.70/– | – | – |
| **BERT** | | | | |
| BERT [19] | BERT context encoder | 87.70/– | – | – |
| BERT+RNN [101] | BERT context encoder+RNN state decoder | 89.20/– | – | – |
| BERT† [98] | BERT context & candidate encoder | 90.50/97.60 | – | – |
| DistilledBERT† [98] | Distilled variant of BERT† | 90.40/**97.70** | – | – |
| SUMBT [101] | BERT+RNN+slot-utterance attention | 91.00/– | – | – |
| **XLM** | | | | |
| XLM-R-DST [261] | XLM-R context encoder with 270M parameters | 88.50/– | – | – |
| XQA-DST [261] | XLM-R+value span extraction | **92.38**/– | – | – |
| **T5 (Ours)** | | | | |
| mT5 | Multilingual T5 with 300M parameters | 89.53/97.02 | 79.06/95.92 | 87.58/95.44 |
| mT5+bMOLR | mT5+bilingual mixture-of-languages routing | 91.42/**97.32** | **81.62**/96.65 | **88.25**/96.53 |
| mT5+mMOLR | mT5+multilingual mixture-of-languages routing | **91.84**/97.02 | 81.56/**97.02** | 87.77/96.41 |

## 4.5.1 Main results (RQ4.1)

We compare the performance of MOLR models with the existing monolingual models and multilingual models on both the DST (see Table 4.3) and the NLU (see Table 4.4) task.

### MOLR improves both monolingual and multilingual DST

From the results for the DST task in Table 4.3, we have the following observations.

First, the mT5 models with MOLR outperform all monolingual and multilingual baselines for German and Italian. They also achieve higher scores than most of the scores reported for English. Specifically, mT5+bMOLR significantly outperforms mT5 by 1.89%/2.56%/0.67% of joint goal accuracy and 0.3%/0.73%/1.09% of request accuracy for English/German/Italian. The improvements prove the effectiveness of MOLR. We believe the main reason is that MOLR is able to explore pairwise relationships between languages and to fully make use of multilingual data for global optimization. Although

XQA-DST and DistilledBERT† achieve slightly higher results than mT5+bMOLR for English in terms of joint goal accuracy (+0.54%) and request accuracy (+0.38%), the predictive space is much smaller than MOLR models. This is mainly because QA-DST and DistilledBERT† are classification models, in contrast, the MOLR models are generation models.

Second, mT5 is the state-of-the-art base model compared with all types of base models. More precisely, mT5 (89.53%) achieves the highest joint goal accuracy for English, followed by XLM-R-DST (88.50%), GLAD (88.10%), BERT (87.70%), NBT (84.20%). mT5 dramatically improves the existing reported results on German and Italian. Specifically, it increases 10.96% and 9.48% over the best NBT results in German and Italian, respectively.

Third, pairwise alignment brings consistent improvement. MOLR improves over mT5 in all settings. However, NBT+mDA decreases the joint goal accuracy by 1.40% for English compared with monolingual NBT. The benefit of multilingual data for training appears to be limited without modeling the language relationships. When adding more languages, mT5+mMOLR achieves slight increases and slight decreases. Compared with mT5+bMOLR, mT5+mMOLR improves the joint goal accuracy by 0.42% and the request accuracy by 0.37%, but slightly drops in the remaining settings.

Fourth, global optimization of multilingual DST is still underexplored. NBT+Morph finetunes German and Italian models with multilingual word embeddings. NBT+mDA uses multilingual data during training for global optimization. However, recent models ignore the performance in German and Italian. This might be because most research is English-centered: either English models or cross-lingual adaptation from English to other languages.

### MOLR improves monolingual and multilingual NLU

From the results on the NLU task in Table 4.4, we have the following observations.

First, MOLR models outperform or are on par with all monolingual and multilingual baselines for English, Spanish, and Thai. Particularly, mT5+bMOLR and mT5+bMOLR improve over mT5 by 5.32%/1.14% and 5.13%/1.75% of slot F1 for Thai and Spanish. The improvements prove the effectiveness of MOLR. The gain of MOLR is limited in other settings, including evaluation results on English and intent accuracy. The general improvement is smaller than 0.5%. One reason is that the volume of data is already sufficient for good intent identification. For example, biLSTM-CRF achieves 99.11% of accuracy on intent identification and 94.81% of slot F1 for English. Thus, the performance of slot filling leaves more room for improvement than intent identification. Another reason is that the extra information from low-resource languages (e.g., Spanish and Thai when used in combination with English) is quite limited. For example, in the NLU dataset, only 11.7% and 20.0% of the utterances are parallel with English, respectively, which is all that is available to help improve the English model. In contrast, the rest of the non-parallel English utterances can bring new information to the low-resource languages.

Second, mT5 is the state-of-the-art base model compared with all types of base models similar to DST (see Table 4.3). Specifically, mT5 (96.40%) obtains the highest slot F1 for English, followed by mBERT (95.97%), transformers (94.93%), and biLSTM

Table 4.4: Comparison of natural language understanding (NLU) models for supervised learning using English, Spanish, and Thai as target languages. In the cells with results, the numbers before and after "/" denote intent accuracy and slot F1 score, respectively. Boldface indicates leading results.

| | | **Intent Accuracy/Slot F1 (%)** | | |
|---|---|---|---|---|
| **NLU Models** | **Settings** | **English** | **Spanish** | **Thai** |
| **RNNs** | | | | |
| biLSTM [121] | biLSTM for only target language | –/94.87 | – | – |
| biLSTM-CRF [190] | Monolingual biLSTM with CRF layer | 99.11/94.81 | 97.26/80.95 | 95.13/87.26 |
| CoVe [190] | biLSTM-CRF based NMT to English | – | 97.81/82.55 | 96.87/90.60 |
| mCoVe [190] | Multilingual CoVe [139] | – | 97.82/82.49 | 96.98/91.22 |
| mCoVe+Auto [190] | mCoVe with autoencoder objective | – | 97.90/82.13 | 96.87/91.51 |
| **Transformers** | | | | |
| Transformer [121] | Transformer w/frozen word embeddings | –/94.93 | – | – |
| **BERT** | | | | |
| mBERT [121] | mBERT fine-tuning | –/95.97 | – | – |
| mBERT+DA [183] | mBERT+ monolingual data augmentation | – | 98.20/84.27 | 91.42/59.68 |
| **XLM** | | | | |
| XLM-R [63] | XLM-R encoder with 270M parameters | – | 98.70/89.10 | 96.80/93.10 |
| XLM-R+TA [63] | XLM-R+translation alignment loss | 99.30/**96.60** | 98.80/89.80 | **97.80**/94.40 |
| **T5 (Ours)** | | | | |
| mT5 | Multilingual T5 with 300M parameters | 99.35/96.40 | 98.68/88.45 | 97.52/89.48 |
| mT5+bMOLR | mT5+bilingual mixture-of-languages routing | 99.29/96.49 | **99.08**/89.59 | 97.28/**94.81** |
| mT5+mMOLR | mT5+multilingual mixture-of-languages routing | **99.40**/96.50 | 98.88/**90.21** | 97.70/94.61 |

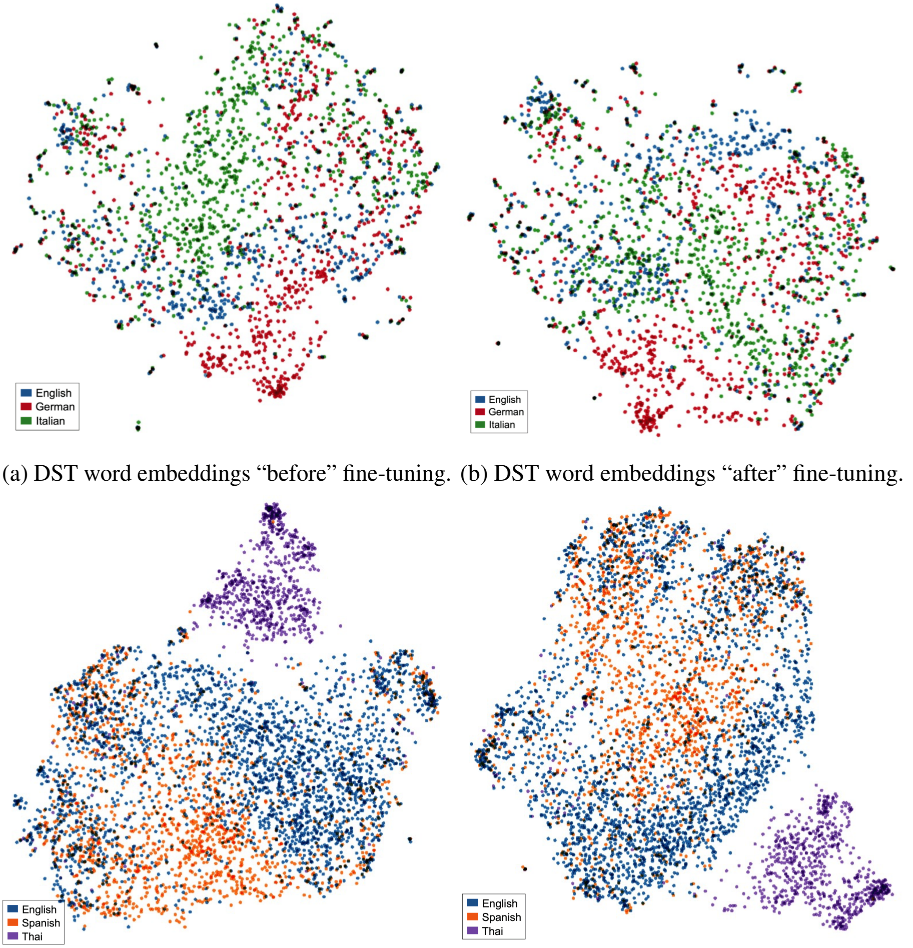
(94.87%). XLM-R is as competitive as mT5, but we choose mT as our backbone considering both the DST performance and the generation benefit (i.e., out-of-ontology prediction).

Third, pairwise alignment brings consistent improvements for slot filling. Compared with mT5, mT5+bMOLR improves 5.32%, 1.14%, and 0.09% of slot F1 for Thai, Spanish and English, respectively. Adding more languages to mT5+bMOLR, mT5+mMOLR brings a small increase for most settings, except for a small decrease in slot F1 for Thai (-0.2%). Similarly, it depends on how much meaningful information a new language can bring to learn better relationships.

Fourth, global optimization of multilingual NLU is still underexplored. Even for language-specific optimization, biLSTM-CRF and XLM-R+TA are the only approaches for which results are reported on all languages, to the best of our knowledge. This might be because most research focuses on cross-lingual adaptation from English to other low-resource languages in NLU dataset.

### 4.5.2 Analysis of language characteristics (RQ4.2)

In Section 4.5.1, we observe that the overall performance varies a lot for different languages. In this section, we first analyze the language characteristics (i.e., unity and diversity) in depth by visualizing the word embeddings, as well as genetic, word, and sentence similarities of different languages. Then we analyze the gains of different



(a) DST word embeddings "before" fine-tuning. (b) DST word embeddings "after" fine-tuning.

(c) NLU word embeddings "before" fine-tuning. (d) NLU word embeddings "after" fine-tuning.

Figure 4.4: Visualization of words in the DST and NLU datasets "before" and "after" fine-tuning. We conduct dimension reduction using the UMAP algorithm [140] and plot all scatter in 2D coordinates using the Tensorflow embedding projector.[9]

languages in different settings.

### Qualitative analysis of the unity and diversity of languages

See Figure 4.4 for visualizations of the word embeddings of mT5+mMOLR in the DST and NLU datasets, before and after fine-tuning, respectively. We aim to understand how MOLR influences the unity and diversity of languages qualitatively.

First, different languages have both similar and dissimilar words in the semantic

[9] https://projector.tensorflow.org/

embedding space. Specifically, some data points from different languages are very close to each other while other data points are far away and located in an isolated cluster. For example, parts of English and German points are mixed up while other sets of German data points are concentrated in an isolated area.

Second, the similarities between languages are very different for different language pairs. For example, the boundaries between English and German, and between English and Italian are not obvious; in contrast, the boundaries between Thai and English, and Thai and Spanish are quite clear. This indicates that English, German, and Italian are quite similar to each other, while Thai is an independent and distinct language that is not similar to English and Spanish. Besides, Thai is closer to English rather than Spanish; the English cluster seems to separate Thai and Spanish.

Third, the relative relationships are not changed before and after fine-tuning. In DST, English, German, and Italian points are mixed together with a small isolated cluster of German points. In NLU, English and Spanish have both shared and non-shared areas, while the majority of Thai points are in an isolated cluster. This shows that English, German, and Italian have more unity, while Thai has more diversity compared with English and Spanish.

### Quantitative analysis of the unity and diversity of languages

As shown in Table 4.5, we evaluate the unity and diversity of languages by three similarity metrics, i.e., genetic similarity, word similarity, and sentence similarity based on word embeddings of mT5 model in the datasets. We aim to quantify the unity and diversity of language and use the similarity order to analyze our MOLR models in the next section.

First, (EN, DE) are the most similar language pair in terms of genetic similarity, followed by (EN, IT), (DE, IT), (EN, ES). Second, (ES, TH) are more similar than (EN, TH) in terms of word and sentence similarity. Last but not least, considering the similarity in one aspect is not always meaningful. For example, the comparison of genetic similarity is invalid for “(EN, TH)” and “(ES, TH)”, and the word similarity of “(EN, DE)” and “(EN, IT)” has very little difference. Another example is that the order of sentence similarity is inconsistent with that of word similarity. Unlike in NLU, the difference in word similarity in DST is small, which might increase the difficulty of distinguishing between sentence embeddings. Hence, it is important to consider all similarity metrics. In this work, we sort the similarity degree of all language pairs in descending order as:

$$\boxed{\phi(EN, DE) > \phi(EN, IT) > \phi(DE, IT) > \phi(EN, ES) > \phi(EN, TH) > \phi(ES, TH)}$$

The point of this order is to fairly compare the similarity between languages; this can be used as language-specific knowledge to analyze how different languages influence MOLR.

### Gains of MOLR are language-specific

We compare mT5 and its variants with different language routes on the DST and NLU tasks, as shown in Table 4.6.

Table 4.5: Similarity between languages. We report genetic similarity, as well as word and sentence similarity based on the DST and NLU datasets.

| | DST: Similarity | | | | NLU: Similarity | | |
|---|---|---|---|---|---|---|---|
| **Language pair** | **Genetic** | **Word** | **Sentence** | **Language pair** | **Genetic** | **Word** | **Sentence** |
| (EN, DE) | 0.1667 | 0.6725 | 0.8813 | (EN, ES) | 0.0833 | 0.7448 | 0.8777 |
| (EN, IT) | 0.1250 | 0.6711 | 0.9036 | (EN, TH) | 0.0000 | 0.4787 | 0.5706 |
| (DE, IT) | 0.0909 | 0.6486 | 0.9066 | (ES, TH) | 0.0000 | 0.4056 | 0.5512 |

Table 4.6: The performance of the proposed mT5-based models with different language routes on the DST and NLU tasks. The bold numbers are the best results in terms of different evaluation metrics for target languages.

| | **DST: Joint Goal / Request Accuracy (%)** | | | | | |
|---|---|---|---|---|---|---|
| **Model** | **English (EN)** | | **German (DE)** | | **Italian (IT)** | |
| mT5 | 89.53/97.02 | | 79.06/95.92 | | 87.58/95.44 | |
| | EN,DE→EN | EN,IT→EN | DE,EN→DE | DE,IT→DE | IT,EN→IT | IT,DE→IT |
| mT5+bMOLR | 91.42/97.32 | 91.11/**97.57** | **81.62**/96.65 | **81.62**/96.23 | **88.25/96.53** | 86.98/96.41 |
| | 1.DE,IT→DE<br>2.EN,DE→EN | 1.IT,DE→IT<br>2.EN,IT→EN | 1.EN,IT→EN<br>2.DE,EN→DE | 1.IT,EN→IT<br>2.DE,IT→DE | 1.EN,DE→EN<br>2.IT,EN→IT | 1.DE,EN→DE<br>2.IT,DE→IT |
| mT5+mMOLR | **91.84**/97.02 | 91.42/97.14 | 81.56/**97.02** | 81.38/96.23 | 87.77/96.41 | 86.00/96.35 |
| | **NLU: Intent Accuracy/ Slot F1 (%)** | | | | | |
| | **English (EN)** | | **Spanish (ES)** | | **Thai (TH)** | |
| mT5 | 99.35/96.40 | | 98.68/88.45 | | 97.52/89.48 | |
| | EN,ES→EN | EN,TH→EN | ES,EN→ES | ES,TH→ES | TH,EN→TH | TH,ES→TH |
| mT5+bMOLR | 99.29/96.49 | 99.29/96.34 | **99.08**/89.59 | 98.78/88.94 | 97.28/94.81 | 97.64/93.39 |
| | 1.ES,TH→ES<br>2.EN,ES→EN | 1.TH,ES→TH<br>2.EN,TH→EN | 1.EN,TH→EN<br>2.ES,EN→ES | 1.TH,EN→TH<br>2.ES,TH→ES | 1.EN,ES→EN<br>2.TH,EN→TH | 1.ES,EN→ES<br>2.TH,ES→TH |
| mT5+mMOLR | **99.40/96.50** | 99.30/96.39 | 98.91/89.53 | 98.88/90.21 | 97.70/94.61 | 97.52/94.59 |

First, gains of MOLR vary when choosing different pivot languages (Section 4.3.3) at different stages of language routes (Section 4.4.3). mT5+bMOLR achieves better performance when the similarity between the source language and pivot language is larger in most settings. Specifically, it obtains 1.35% higher joint goal accuracy for Italian DST using English rather than German as the pivot language. Besides, it gets 1.45% improvements in slot F1 for Thai NLU using English rather than Spanish as the pivot language. For mT5+mMOLR, the first stage is pre-training, and the second stage is the main procedure. It achieves better performance when the similarity between the source language and the pivot language in the second stage is larger in most settings. For example, it improves 1.77% of joint goal accuracy for Italian DST using English rather than German as the pivot language in the second stage. Language transfer is easier if the source and pivot languages are more similar, and it can avoid introducing too many language gaps in the early stage of a language route.

Table 4.7: Comparison of the effect of different combination policies on mT5+bMOLR model. The bold numbers are the best results in terms of different evaluation metrics for target languages.

| | **DST: Joint Goal Accuracy / Request Accuracy (%)** | | | | | |
|---|---|---|---|---|---|---|
| **Model** | **English (EN)** | | **German (DE)** | | **Italian (IT)** | |
| mT5 | 89.53/97.02 | | 79.06/95.92 | | 87.58/95.44 | |
| | EN,DE→EN | EN,IT→EN | DE,EN→DE | DE,IT→DE | IT,EN→IT | IT,DE→IT |
| w/ bDA | 89.59/96.71 | 91.05/96.90 | 79.98/96.35 | 81.25/96.53 | 85.82/**96.71** | 87.89/96.17 |
| w/ route-addressing | 89.11/97.14 | 89.17/95.92 | 79.12/95.44 | 77.97/95.74 | 84.97/95.86 | 80.85/95.01 |
| w/ parameter-sharing | **91.42**/97.32 | 91.11/**97.57** | **81.62/96.65** | **81.62**/96.23 | **88.25**/96.53 | 86.98/96.41 |
| | **NLU: Intent Accuracy / Slot F1 (%)** | | | | | |
| | **English (EN)** | | **Spanish (ES)** | | **Thai (TH)** | |
| mT5 | 99.35/96.40 | | 98.68/88.45 | | 97.52/89.48 | |
| | EN,ES→EN | EN,TH→EN | ES,EN→ES | ES,TH→ES | TH,EN→TH | TH,ES→TH |
| w/ bDA | **99.34/96.53** | 99.32/96.47 | 98.98/89.92 | 98.72/88.57 | **97.87**/94.45 | 97.52/92.06 |
| w/ route-addressing | 99.25/95.93 | 99.22/95.86 | 98.72/**89.89** | 98.45/87.19 | 97.52/93.67 | 97.16/91.33 |
| w/ parameter-sharing | 99.29/96.49 | 99.29/96.34 | **99.08**/89.59 | 98.78/88.94 | 97.28/**94.81** | 97.64/**93.39** |

Second, compared with mT5+bMOLR, the performance of mT5+mMOLR varies with different additional languages in the second stage. For example, mT5+mMOLR increases 0.42% in joint goal accuracy by adding Italian into the additional route of "DE,IT→DE". However, mT5+mMOLR decreases 0.98% in joint goal accuracy by adding English into the additional route of "DE,EN→DE". The second stage is essential, and it is helpful if the source language and pivot language are similar in the second stage.

Third, the performance of mT5+mMOLR is dependent on the volume of additional languages and the difficulty of tasks. For most settings, the changes are small compared with mT5+bMOLR. Particularly, mT5+mMOLR increases 1.27% of slot F1 by adding English into the additional route of "TH,EN→TH" for Spanish NLU. Similarly, mT5+mMOLR increases 1.20% of slot F1 by adding English into the additional route of "ES,EN→ES" for Thai NLU. One reason is that English is a high-resource language compared with Spanish and Thai in NLU, which is able to provide sufficient extra information for improvement. Another reason is that the slot filling task is more difficult than intent identification, and the potential for improvement of the former is larger than the latter.

## 4.5.3 Analysis of key components of mT5+bMOLR (RQ4.3)

### Combination policies are essential

We compare mT5 with its variants, i.e., mT5+bDA which is mT5 with bilingual data training, route-addressing, and parameter-sharing combination policies, as shown in Table 4.7.

First, mT5 with parameter-sharing outperforms mT5 in all settings. Specifically, it improves 2.56%/0.73%, 1.83%/0.61%, and 0.19%/0.96% for German, English, and Italian DST in terms of joint goal accuracy and request accuracy, respectively. Meanwhile, it improves 0.18%/5.13%, 0.23%/1.76%, and 0.05%/0.10% for Thai, Spanish, and English NLU, respectively. This proves the overall effectiveness of MOLR models.

Second, mT5 with parameter-sharing outperforms or is on par with bDA (i.e., bilingual data augmentation) in all settings. In DST, "EN, DE→EN", "EN, DE→DE", "IT,EN→DE" changes +1.83%/+0.61%, +1.64%/+0.3%, +2.43%/-0.17% in terms of joint goal accuracy and request accuracy, given a pivot language similar to source language. However, mT5+bDA cannot always benefit from multilingual data, e.g., "IT,EN→IT" decreases 1.76% of joint goal accuracy compared with mT5. In NLU, "TH,ES→TH" and "ES, TH→ES" mutually increase as much as 2.53% and 1.64% in terms of slot F1, while the changes are quite small (<0.40%) for the other settings. This reveals that the gains are also from MOLR and global optimization, along with multilingual data.

Third, policy parameter-sharing outperforms route-addressing in general. In DST, parameter-sharing beats route-addressing by 1.94%–6.13% and 0.18%–1.64% in terms of joint goal accuracy and request accuracy. In NLU, parameter-sharing outperforms route-addressing by 0.04%–2.40% and 0.48%–2.06% in terms of joint goal accuracy and request accuracy, excluding the intent accuracy for "TH,EN→TH" (-0.24%) and slot F1 for "ES,EN→ES" (-0.30%). Thus, we use parameter-sharing as the combination policy for our best-performing models.

### Impact of the number of layers varies with tasks and languages

We study the influence of the different number of layers for each expert model in Table 4.8.

First, the best settings of layers for expert agents (Section 4.3.2) vary for different tasks. In DST, joint goal accuracy notably decreases 3.59%–6.27%, and request accuracy only decreases by 0.17%–1.59% with reducing the number of layers from 8 to 2. In NLU, slot F1 dramatically reduces by 2.96%–24.23%, in contrast, intent accuracy reduces or even increases slightly, e.g., "TH,EN→TH" improves 0.36%. The difficulty of different tasks varies, and the number of layers has less influence on simpler tasks.

Second, the influence of the number of layers is language-specific. In DST, we reduce the number of layers from 8 to 2. The mixture of German and Italian (i.e., "DE,IT→DE" and "IT,DE→IT") does not always drop like the rest of the settings. Since the amount of multilingual data is comparable, it is likely caused by the language specification, i.e., (DE, IT) are less similar than (EN, DE) and (EN, IT), and the mixture of German and Italian can preserve more diversity.

Third, the number of layers is sensitive to high-resource pivot languages. Reducing the number of layers from 8 to 4, the changes in "EN,ES→EN " and "EN,TH→EN" are less than 0.1% and 0.5% in terms of intent accuracy and slot F1. The pivot languages (i.e., Spanish and Thai) have much fewer data samples compared with the high-resource language, i.e., English.

Table 4.8: Model complexity by agent models with the different number of layers.

| | **DST: Joint Goal Accuracy / Request Accuracy (%)** | | | | | |
|---|---|---|---|---|---|---|
| | **English (EN)** | | **German (DE)** | | **Italian (IT)** | |
| **#Layers** | EN,DE→EN | EN,IT→EN | DE,EN→DE | DE,IT→DE | IT,EN→IT | IT,DE→IT |
| 8 | **91.42/97.32** | **91.11/97.57** | **81.62/96.65** | 81.62/96.23 | **88.25/96.53** | 86.98/**96.41** |
| 6 | 90.75/97.20 | 90.44/97.14 | 80.83/95.01 | **82.47**/95.98 | 85.51/96.10 | **87.16/96.41** |
| 4 | 88.92/97.20 | 88.98/97.02 | 78.76/96.04 | 79.79/**96.53** | 85.09/96.35 | 83.99/95.98 |
| 2 | 86.49/97.14 | 86.79/97.02 | 76.38/95.56 | 78.03/94.64 | 81.98/95.19 | 81.56/95.19 |
| | **NLU: Intent Accuracy / Slot F1 (%)** | | | | | |
| | **English (EN)** | | **Spanish (ES)** | | **Thai (TH)** | |
| | EN,ES→EN | EN,TH→EN | ES,EN→ES | ES,TH→ES | TH,EN→TH | TH,ES→TH |
| 8 | 99.29/**96.49** | 99.29/96.34 | **99.08**/89.59 | **98.78/88.94** | 97.28/**94.81** | 97.64/**93.39** |
| 6 | **99.38**/96.37 | 99.27/96.33 | 98.88/**89.87** | 98.68/88.33 | **97.58**/93.27 | **97.66**/91.58 |
| 4 | 99.28/96.07 | 99.27/96.01 | 98.95/87.28 | 98.55/83.75 | 97.40/91.61 | 97.22/84.87 |
| 2 | 99.30/93.53 | **99.33**/93.68 | 98.62/83.59 | 98.39/66.25 | 97.64/84.82 | 96.93/69.16 |

## 4.6 Conclusion and future work

In this chapter, we have studied multilingual TDSs in a collaborative TDS framework, where expert agents work on monolingual and cross-lingual dialogues, and the chair agent accounts for a mixture-of-experts approach for globally optimizing multilingual dialogues. We have proposed a mixture-of-languages routing (MOLR) paradigm, which aims to fully make use of multilingual data, capture language relationships, and globally optimize multilingual performance simultaneously. We have conducted experiments on two benchmark multilingual TDS datasets to verify the effectiveness of the proposed MOLR based on a pre-trained mT5 model.

Our main finding is that MOLR can be greatly influenced by data availability, language characteristics, as well as collaboration policies. To be precise, training MOLR with sufficient multilingual data can significantly improve performance over training with little data in a low-resource language. Moreover, MOLR with increasing amounts of data in different languages can perform very differently, so the gains of MOLR are language-specific. Different combination policies enable global optimization, and their performance varies a lot; this demonstrates the versatility and effectiveness of the collaborative paradigm. Together, these findings and insights provide an affirmative answer to the leading research question for this chapter: multiple languages can indeed be used in a collaborative way to improve the performance of task-oriented dialogue systems in every single language.

As to broader implications of multilingual TDSs, researchers in this domain should consider as many languages as possible. They should also enable their models to select valuable data for enhancement. Language characteristics (e.g., unity and diversity) should never be underestimated.

One limitation of this work is that MOLR can only work when multilingual data can bring both commonalities and peculiarities of languages: In the extreme case where two languages do not have any commonalities, MOLR can hardly learn to transfer across languages for cross-lingual adaptation. And vice versa, if two languages do not have any peculiarities, MOLR can hardly gain from language transfer for a multilingual model.

As to future work, we believe that multilingual TDSs can be advanced in many directions. First, we plan to use different pre-trained language models (e.g., GPT2, mBART, etc.) and compare them with the mT5 model. Second, we plan to explore different collaboration policies and see how they influence overall performance. Third, we plan to experiment on new datasets with full dialogue tasks and more languages and step forward to practical applications of multilingual TDSs.

Next, we switch from collaboration to uncertainty estimation for collaborative agents.

# Appendix

## 4.A Summary of zero-shot cross-lingual benchmarks

We summarize all the zero-shot crosslingual results on the DST (Table 4.9) and NLU (Table 4.10) datasets, as well as our implementation of mT5 models. We find that mT5 and its variants achieve the state-of-the-art for zero-shot crosslingual adaptation. This justifies our choice of mT5 as our base model in the main results.

Table 4.9: Comparison of dialogue state tracking (DST) models for zero-shot learning from English (EN) to German (DE) and Italian (IT).

| | | **Joint/Request (%)** | |
|---|---|---|---|
| **Models** | **Settings** | **EN → DE** | **EN → IT** |
| **NBT** | | | |
| XL-NBT [23] (from [120]) | Teacher-student NBT+bilingual data augmentation | 30.80/68.32 | 41.23/81.23 |
| **MUSE** | | | |
| MUSE [120] | Word alignment using MUSE [27] | 21.57/74.22 | 20.66/79.09 |
| MUSE+AMLT [120] | MUSE+attention-based bilingual code-switching | 36.51/82.99 | 39.35/84.23 |
| **XLM** | | | |
| XLM [120] | XLM [26] context encoder | 16.34/75.73 | – |
| XLM+AMLT [120] | XLM+attention-based bilingual code-switching | 33.12/82.96 | – |
| XLM+CLCSA [174] | XLM+multilingual code-switching | 48.70/88.30 | – |
| XQA-DST [261] | XLM-R [28]+value span extraction | 64.88/– | 68.63/– |
| **BERT** | | | |
| mBERT [120] | mBERT [39] context encoder | 14.95/75.31 | 12.88/76.12 |
| mBERT+AMLT [120] | mBERT+attention-based bilingual code-switching | 34.36/86.97 | 33.35/84.96 |
| mBERT+CLCSA [174] | mBERT+multilingual code-switching | 63.20/94.00 | 61.30/**94.20** |
| **T5 (Ours)** | | | |
| mT5 | Multilingual T5 (small) with 300M parameters | 28.42/92.27 | 32.14/87.22 |
| mT5+AMLT | mT5+bilingual code-switching | 40.96/93.37 | 47.90/87.65 |
| mT5+CLCSA | mT5+multilingual code-switching | **67.86/95.80** | **71.15**/88.07 |

Table 4.10: Comparison of natural language understanding (NLU) models for zero-shot learning from English (EN) to German (DE) and Italian (IT).

| | | **Intent/Slot F1 (%)** | |
|---|---|---|---|
| **Models** | **Settings** | **EN → ES** | **EN → TH** |
| **RNN** | | | |
| biRNN [119] | An implantation of bidirectional RNN [213] | 46.64/15.41 | 35.64/12.11 |
| CoVe [190] | biLSTM-CRF based translation model to English | 37.13/ 5.35 | 54.24/ 8.84 |
| mCoVe [190] | Multilingual CoVe [139] | 53.34/22.50 | 66.35/32.52 |
| mCoVe+Auto [190] | mCoVe w/autoencoder objective | 53.89/19.25 | 70.70/35.62 |
| biLSTM [119] | biLSTM w/noise, refinement, delexicalization | 90.20/65.79 | 73.43/32.24 |
| **MUSE** | | | |
| RCSLS [86] | MUSE+relaxed cross-domain similarity local scaling | 37.67/22.23 | 35.12/ 8.72 |
| RCSLS+AMLT [120] | RCSLS+attention-based bilingual code-switching | 87.05/57.75 | 81.44/30.42 |
| **Transformers** | | | |
| Transformer [121] | Transformer w/frozen word embeddings | 89.71/67.10 | 74.68/31.20 |
| Transformer+ORT [121] | Order-reduced transformer | 91.46/71.36 | 75.02/34.61 |
| **mBERT** | | | |
| mBERT [120] | mBERT [39] context encoder | 74.15/54.28 | 26.54/11.34 |
| mBERT+AMLT [120] | mBERT+attention-based bilingual code-switching | 87.88/73.89 | 73.46/27.12 |
| mBERT+CLCSA [174] | mBERT+multilingual code-switching | 92.80/75.20 | 74.80/28.10 |
| XLM-R [63] | XLM-R encoder with 270M parameters | 90.70/70.10 | 71.90/53.10 |
| **T5 (Ours)** | | | |
| mT5 | Multilingual T5 with 300M parameters | 92.17/71.26 | 81.38/52.13 |
| mT5+AMLT | mT5+bilingual code-switching | 92.77/71.92 | 91.61/**57.46** |
| mT5+CLCSA | mT5+multilingual code-switching | **94.71/75.77** | **93.20**/47.02 |

# 5
# Uncertainty Estimation: Hierarchical Stochastic Transformer

In this chapter, we aim to answer the following research question:

**RQ4** Can we enable collaborative agents with the capability of uncertainty estimation towards trustworthy systems?

We rethink vanilla transformer as a sequence of collaborative agents, which is stacked with a sequence of blocks and serves as the backbone of many state-of-the-art models. We propose two variants of vanilla transformer: (i) STO-TRANS, which injects stochasticity into the stochastic attention over values; and (ii) H-STO-TRANS as an extension, which forces key heads to pay stochastic attention to a set of learnable centroids. Our main finding is that the proposed models outperform compared models and enable us to trade off the performance between in-domain (ID) prediction and out-of-domain (OOD) uncertainty estimation on three benchmark tasks, i.e., sentiment analysis (SA), linguistic acceptability (LA) and slot filling (SF).

## 5.1 Introduction

Uncertainty estimation and quantification are important tools for building trustworthy and reliable machine learning systems [87, 110, 186]. Particularly, when such machine-learned systems are applied to make predictions that involve important decisions, e.g., medical diagnosis [59], financial planning and decision-making  [6, 157], and autonomous driving [74]. The recent development of neural networks has shown excellent predictive performance in many domains. Among those, transformers, including the vanilla transformer [216] and its variants such as BERT [39, 220], are the representative state-of-the-art type of neural architectures that have shown remarkable performance on various natural language processing (NLP) [60] and information retrieval (IR) [185] tasks.

Although transformers excel in terms of predictive performance [66, 209], they do not offer the opportunity for practitioners to inspect the model confidence due to their

This chapter was published as J. Pei, C. Wang, and G. Szarvas. Transformer uncertainty estimation with hierarchical stochastic attention. In *AAAI*, pages 11147–11155, 2022 [170].

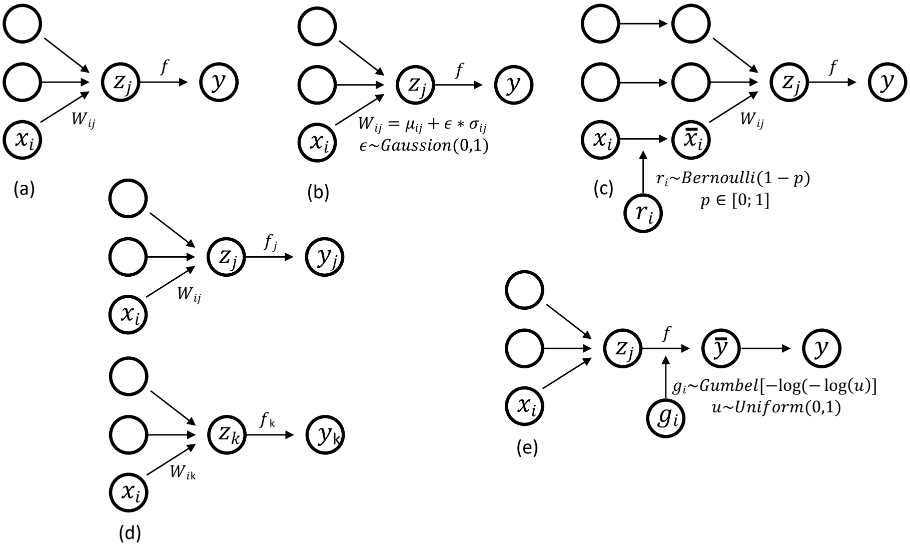


Figure 5.1: Methods for uncertainty estimation. (a) A deterministic neural network outputs a single-point prediction; (b) A Bayesian neural network captures uncertainty by sampling from a Gaussian distribution; (c) Variational dropout captures uncertainty by sampling dropout masks from a Bernoulli distribution; (d) An ensemble captures uncertainty by combining multiple independently trained deterministic models with different random seeds; and (e) The Gumbel-softmax trick for uncertainty estimation, where the randomness comes from sampling the categorical distribution from a Gumbel distribution.

deterministic nature, i.e., it is not possible to assess if transformers are confident about their predictions. This influence is non-trivial because transformers are cutting-edge basic models for NLP. Thus, estimating the predictive uncertainty of transformers potentially benefits a range of applications in terms of building and examining model reliability for downstream tasks.

To estimate the uncertainty of neural models' prediction, one common way is to inject stochasticity (e.g., noise or randomness) [58, 87]. This enables models to output a predictive distribution instead of a single-point prediction. Casting a deterministic transformer to be stochastic requires us to take the computational complexity of training and inference into consideration because uncertainty estimation usually relies on multiple forward runs. Therefore, directly adapting the aforementioned methods is not desired, given the large number of parameters and architectural complexity of transformers.

Figure 5.1 outlines the deterministic transformer (Figure 5.1(a)) and the possible approaches (Figure 5.1(b-e) for making a stochastic transformer. Bayesian neural network (BNN) (Figure 5.1(b)) assumes the network weights follow a Gaussian or a mixture of Gaussian [11], and tries to learn the weight distribution $(\mu, \sigma)$, instead of weight $W$ itself, with the help of re-parameterization trick [92]. This means that BNN doubles the number of parameters. This is particularly challenging for a large network like a transformer, which has millions of parameters to be optimized. To alleviate this issue, MC dropout [54] (Figure 5.1(c)) uses dropout [204], concretely Bernoulli distributed random variables, to approximate the exact posterior distribution [54]. However, MC dropout tends to give overconfident uncertainty estimations [51]. An ensemble [99] (Figure 5.1(d)) is an alternative way to model uncertainty by averaging $N$ independently trained models, which yields a computational overhead by $N$ times in model training.

Unlike the models above, we propose a simple yet effective approach, based on Gumbel-softmax tricks or concrete dropout [81, 132], to estimate the uncertainty of transformers. Gumbel-softmax tricks are proposed specifically for continuous relaxation of discrete distributions [172]. First, we cast the deterministic attention distribution for values in each self-attention head to be stochastic. The attention is then sampled from a Gumbel-softmax distribution, which controls the concentration over values.

Second, we regularize the key heads in self-attention to attend to a set of learnable centroids. This is equivalent to performing clustering over keys [219] or clustering hidden states in RNN [222, 223]. A similar attention mechanism has also been used to allow the layers in the encoder and decoder to attend to inputs in the set transformer [102] and to estimate attentive matrices in Capsule networks [1].

Third, each new key head will be formed with a mixture of Gumbel-softmax sampled centroids. Stochasticity is injected by sampling from a Gumbel-softmax distribution. This is different from a BNN (sampling from a Gaussian distribution), MC-dropout (sampling from a Bernoulli distribution), and an ensemble (the stochasticity comes from random seeds in model training).

With this proposed mechanism, we approximate the vanilla transformer with a stochastic transformer based on a hierarchical stochastic self-attention, namely H-STO-TRANS, which enables the sampling of attention distributions over values as well as over a set of learnable centroids.

Our work in this chapter makes the following contributions:

- We propose a novel way to cast self-attention in transformers to be stochastic, which enables transformer models to provide uncertainty information with their predictions.

- We theoretically show that the proposed self-attention approximation is upper bounded; the key attention heads that are close in Euclidean distance have similar attention distribution over centroids.

- In three benchmark tasks for NLP, we empirically demonstrate that H-STO-TRANS (i) achieves very competitive (in most cases, better) predictive performance on in-domain datasets; (ii) is on par with baselines in uncertainty estimation on out-of-domain datasets; and (iii) learns a better predictive performance-uncertainty trade-off than compared baselines, i.e., high predictive performance and low uncertainty on in-domain datasets, high predictive performance and high uncertainty on out-of-domain datasets.

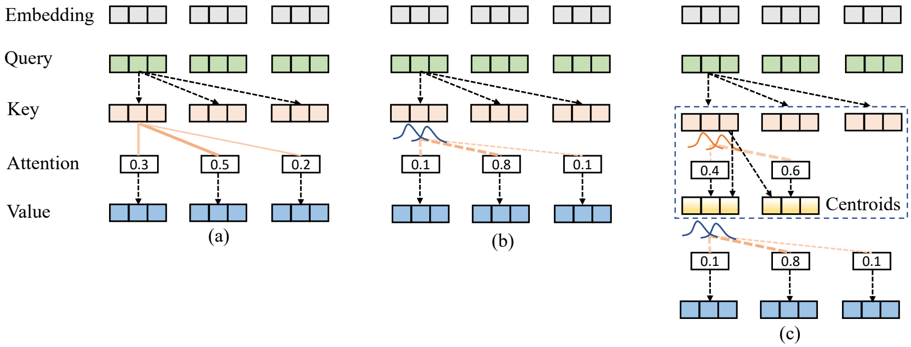


Figure 5.2: Illustration of multi-head self-attention in deterministic and stochastic transformers. (a) The vanilla transformer with deterministic self-attention. (b) The stochastic transformer has stochastic self-attention used to weight values $V$; the standard softmax is replaced with the Gumbel-softmax. (c) The hierarchical stochastic transformer learns to pay attention to values $V$ and a set of learnable centroids $C$ stochastically.

## 5.2 Related work

In this section, we summarize three classes of mainstream approaches for uncertainty estimation in neural networks: (i) Bayesian neural networks, (ii) deep ensembles, and (iii) neural networks based on MC dropout. The discussion is organized as follows. For each class of approach, we explain why stochasticity can be introduced and the drawback. Then we compare our proposed models with the mainstream approaches.

Bayesian neural networks [11] inject stochasticity by sampling the network parameters from a Gaussian prior. Then the posterior distribution of the target can be estimated in multiple sampling runs. However, the Bayesian approach doubles the number of network parameters, i.e., instead of learning a single-point network parameter, it learns a weight distribution that is assumed to follow a Gaussian distribution. Additionally, it often requires intensive tuning on Gaussian mean and variance to achieve stable learning curves as well as predictive performance.

Deep ensembles [99] alternatively offer the possibility to estimate predictive uncertainty by combining predictions from different models which are trained with different random seeds. This, however, significantly increases the computational overhead for training and inference.

Many recent works have proposed various models based on MC dropout. Sequential MC transformer [135], models uncertainty by casting self-attention parameters as unobserved latent states. He et al. [68] combined mix-up, self-ensembling, and dropout to achieve more accurate uncertainty scores for text classification. Shelmanov et al. [196] proposed to incorporate determinantal point process (DPP) to MC dropout to quantify the uncertainty of transformers. To sum up, MC dropout [54] approximates the Bayesian approach by sampling dropout masks from a Bernoulli distribution. However, MC dropout has been demonstrated to give overconfident uncertainty estimation [51].

Different from the above-mentioned approaches, we inject stochasticity into the vanilla transformer with Gumbel-softmax tricks. As we show in the experimental results section in this chapter, the hierarchical stochastic self-attention component can effectively capture model uncertainty and learn a good trade-off between in-domain predictive performance and out-of-domain uncertainty estimation.

## 5.3 Background

### 5.3.1 Predictive uncertainty

Predictive uncertainty estimation is a challenging and unsolved problem. It has many faces, depending on different classification rules. Commonly, it is classified as epistemic (model) or aleatoric (data) uncertainty [38, 89]. Alternatively, on the basis of the input data domain, it can also be classified into *in-domain* (ID) [4] and *out-of-domain* (OOD). uncertainty [72, 226]. With in-domain data, i.e., the input data distribution is similar to the training data distribution, a reliable model should exhibit high predictive performance (e.g., high accuracy or F1-score) and report high confidence (low uncertainty) on correct predictions. In contrast, out-of-domain data may have quite different distributions from the training data; an ideal model should give a high predictive performance to illustrate the generalization to unseen data distribution but desired to be unconfident (high uncertainty). We discuss the epistemic (model) uncertainty in the context of ID and OOD scenarios in this chapter.

### 5.3.2 Vanilla transformer

The vanilla transformer [216] is an alternative architecture to recurrent neural networks (RNNs) for modeling sequential data that relaxes the models reliance on input sequence order. It consists of multiple components such as positional embedding, residual connection, and multi-head scaled dot-product attention. The core component of the transformer is the multi-head self-attention mechanism.

Let $\mathbf{x} \in \mathbb{R}^{l \times d}$ be input data, and $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times d}$ be the matrices for query $Q \in \mathbb{R}^{l \times h \times d_h}$, key $K \in \mathbb{R}^{l \times h \times d_h}$, and value $V \in \mathbb{R}^{l \times h \times d_h}$, where $l$ is sequence length, $d$ is dimension, $d_h = \frac{d}{h}$, and $h$ is the number of attention heads. Each $\mathbf{x}$ is associated with a query $Q$ and a key-value pair $(K, V)$. The computation of an attentive representation $A$ of $\mathbf{x}$ in the multi-head self-attention is:

$$Q = \mathbf{W}_q\mathbf{x},\ K = \mathbf{W}_k\mathbf{x},\ V = \mathbf{W}_v\mathbf{x}, \tag{5.1}$$

$$A = \mathrm{softmax}(\alpha^{-1}QK^\top),\ H = AV, \tag{5.2}$$

where $H = [h_1, \dots, h_h]$ is the multi-head output and $A = [a_1, \dots, a_h]$ is the attention distribution that needs to attend to $V$, $\alpha$ is a scaling factor. Note that a large value of $\alpha$ pushes the softmax function into regions where it has extremely small gradients. This attention mechanism is the key factor of a transformer for achieving high computational efficiency and excellent predictive performance. However, as we can see, all computation paths in this self-attention mechanism are deterministic, leading to a single-point output. This limits us from accessing and evaluating the uncertainty information beyond a single prediction given an input $\mathbf{x}$.

We argue that being able to examine the reliability and confidence of a transformer prediction is crucial for many NLP applications, particularly when the output of a model is directly used to serve customer requests. In the following section, we introduce a simple yet efficient way to cast the deterministic attention to be stochastic for uncertainty estimation based on Gumbel-softmax tricks [81, 132].

## 5.4 Methodology

### 5.4.1 Bayesian inference and uncertainty modeling

In this chapter, we focus on using transformers in classification tasks. Let $D = \{X, Y\} = \{x_i, y_i\}_{i=1}^{N}$ be a training dataset, $y_i \in \{1, \ldots, M\}$ is the categorical label for an input $x_i \in \mathbb{R}^d$. The goal is to learn a transformation function $f$, which is parameterized by weights $\omega$ and maps a given input $x$ to a categorical distribution $y$. The learning objective is to minimize the negative log likelihood, $\mathcal{L} = -\frac{1}{N}\sum_i^N \log p(y_i|x_i, \omega)$. The probability distribution is obtained by a softmax function as:

$$p(y_i = m|x_i, \omega) = \frac{\exp(f_m(x_i, \omega))}{\sum_{k\in M}\exp(f_k(x_i, \omega)}. \tag{5.3}$$

During the inference phase, given a test sample $x^*$, the predictive probability $y^*$ is computed by:

$$p(y^*|x^*, D) = \int p(y^*|x^*, \omega)p(\omega|D)d\omega, \tag{5.4}$$

where the posterior $p(\omega|D)$ is intractable and cannot be computed analytically. A variational posterior distribution $q_\theta(\omega)$, where $\theta$ are the variational parameters, is used to approximate the true posterior distribution by minimizing the Kullback-Leilber (KL) distance. It can also be treated as the maximization of the evidence lower bound (ELBO):

$$\mathcal{L}_\theta = \int q_\theta(\omega)p(Y|X, \omega)d\omega - \mathrm{KL}[q_\theta(\omega) \parallel p(\omega)]. \tag{5.5}$$

With the re-parametrization trick [93], a differentiable mini-batched Monte Carlo estimator can be obtained.

The predictive (epistemic) uncertainty can be measured by performing $T$ inference runs and averaging the predictions:

$$p(y*|x*) = \frac{1}{T}\sum_{t=1}^{T} p_{\omega_t}(y^*|x^*, \omega_t). \tag{5.6}$$

$T$ corresponds to the number of sets of mask vectors from the Bernoulli distribution $\{r^t\}_{t=1}^{T}$ in MC-dropout, or the number of randomly trained models in Ensemble, which potentially leads to different set of learned parameters $\omega = \{\omega_1, \ldots, \omega_t\}$, or the number of sets of sampled attention distribution from Gumbel distribution $\{g^t\}_{t=1}^{T}$ in our proposed method.

### 5.4.2 Stochastic self-attention with Gumbel-Softmax

As described in Section 5.3.2, the core component that makes a transformer successful is the multi-head self-attention. For each $i$-th head, let $q_i \in Q, k_i \in K, v_i \in V$, then:

$$a_i = \text{softmax}\left(\frac{q_i k_i^\top}{\tau}\right), a_i \in \mathbb{R}^{l \times l}, \tag{5.7}$$

$$h_i = a_i v_i, h_i \in \mathbb{R}^{l \times d_h}. \tag{5.8}$$

Here we use a temperature parameter $\tau$ to replace the scaling factor $\alpha$. The $a_i$ is the attention distribution, which learns the compatibility scores between tokens in the sequence with the $i$-th attention head. The scores are used to retrieve and form a mixture of the content of values, which is a kind of content-based addressing mechanism in a neural Turing machine [62]. Note that the attention is deterministic.

A straightforward way to inject stochasticity is to replace standard Softmax with Gumbel-Softmax, which helps to sample attention weights to form $\hat{a}_i$:

$$\hat{a}_i \sim \mathcal{G}\left(\frac{q_i k_i^\top}{\tau}\right), \tag{5.9}$$

$$h_i = \hat{a}_i v_i, \tag{5.10}$$

where $\mathcal{G}$ is the Gumbel-softmax function. The Gumbel-softmax trick is an instance of a path-wise Monte-Carlo gradient estimator [64, 81, 132]. With the Gumbel trick, we can draw samples $z$ from a categorical distribution given by parameters $\boldsymbol{\theta}$, that is,

$$\boldsymbol{z} = \text{ONE\_HOT}\big(\text{argmax}_i[g_i + \log \theta_i]\big), i \in \{1, \ldots, k\},$$

where $k$ is the number of categories and $g_i$ are i.i.d. samples from the $\text{GUMBEL}(0, 1)$, that is, $g = -\log(-\log(u)), u \sim \text{UNIFORM}(0, 1)$ is independent of network parameters. Because the $\text{argmax}$ operator breaks end-to-end differentiability, the categorical distribution $\boldsymbol{z}$ can be approximated using the differentiable softmax function [81, 132]. Here, $\tau$ is a tunable temperature parameter equivalent to $\alpha$ in Eq. 5.2. Then the attention weights (scores) for values in Eq. 5.2 can be computed as:

$$\hat{a}_i = \frac{\exp((\log(\theta_i) + g_i)/\tau)}{\sum_{j=1}^{k} \exp((\log(\theta_{t_j}) + g_j)/\tau)}, i \in \{1, \ldots, k\}. \tag{5.11}$$

where $\theta_i = q_i k_i^\top$. And we use the following approximation:

$$\text{KL}[a \parallel \hat{a}] \text{ where } a_j = \frac{a_j}{\sum_k^{i=1} a_i}. \tag{5.12}$$

This indicates an approximation of a deterministic attention distribution $a$ with a stochastic attention distribution $\hat{a}$. With a larger $\tau$, the distribution of attention is more uniform, and with a smaller $\tau$, the attention becomes more sparse.

### The trade-off between predictive performance and uncertainty estimation

This trade-off is rooted in the familiar bias-variance trade-off. Let $\phi(x)$ be a prediction function, and $f(x)$ is the true function and $\rho$ is a constant number. The error can be computed as:

$$\xi(x) = \underbrace{(\mathbb{E}[\phi(x) - f(x)])^2}_{Bias^2} + \underbrace{(\mathbb{E}[\phi(x) - E[\phi(x)]]^2)}_{Variance} + \underbrace{\rho}_{Const} . \tag{5.13}$$

MC-dropout [54] with $T$ times Monte Carlo estimation gives a prediction $\mathbb{E}[\phi_t(x)], t \in T$ and predictive uncertainty, e.g., variances $Variance[\phi_t(x)]$ ($\rho$ is a constant number that denotes irreducible error). On both in-domain and out-of-domain datasets, a good model should exhibit low bias, which ensures model generalization capability and high predictive performance. For epistemic (model) uncertainty, we expect the model to output low variance on in-domain data and high variance on out-of-domain data.

We empirically observe (from Table 5.1 and Table 5.2) that this simple modification in Eq. 5.9 can effectively capture the model uncertainty, but it struggles to learn a good trade-off between predictive performance and uncertainty estimation. That is, when good uncertainty estimation performance is achieved on out-of-domain data, the predictive performance on in-domain data degrades. To address this issue, we propose a hierarchical stochastic self-attention mechanism.

## 5.4.3 Hierarchical stochastic self-attention

To further encourage a transformer model to have stochasticity and retain predictive performance, we propose to add additional stochastic attention before the attention that pays values. This attention forces each key head stochastically to attend to a set of learnable centroids, which will be learned during back-propagation. This is equivalent to regularizing key attention heads. Similar ideas have been used to improve transformer efficiency [219] and to improve RNN memorization [222].

We first define the set of $c$ centroids, $C \in \mathbb{R}^{d_h \times c}$. Let each centroid $c_i \in \mathbb{R}^{d_h}$ have the same dimension as each key head $k_j \in \mathbb{R}^{d_h}$. The model will first learn to pay attention to centroids, and a new key head $\hat{k}_j$ is formed by weighting each centroid. Then $\hat{k}$ and a query $q$ decides the attention weights to combine values $v$. For the $i$-th head, a given query $q_i$, key $k_i$, value $v_i$, the stochastic self-attention can be hierarchically formulated as:

$$\hat{a}_c \sim \mathcal{G}(\tau_1^{-1} k_i C), \hat{a}_c \in \mathbb{R}^{l \times c}, \tag{5.14}$$

$$\hat{k}_i = \hat{a}_c C^\top, \hat{k}_i \in \mathbb{R}^{l \times d_h}, \tag{5.15}$$

$$\hat{a}_v \sim \mathcal{G}(\tau_2^{-1} q_i \hat{k}_i^\top), \hat{a}_v \in \mathbb{R}^{l \times l}, \tag{5.16}$$

$$h_i = \hat{a}_v v_i. \tag{5.17}$$

Here, $\hat{a}_c, \hat{a}_v$ are the sampled categorical distributions that are used to weight centroids in $C$ and tokens in $v_i$. The $\tau_1, \tau_2$ control the softness for each stochastic self-attention, respectively.

We summarize the main procedures of performing hierarchical stochastic attention in the transformer in Algorithm 2.

**Algorithm 2:** Hierarchical stochastic transformer.

```
Input    : Query $Q$, key $K$, value $V$, centroids $C$
Output   : Hierarchical stochastic attentive output $H$
1 Model stochastic attention $\hat{A}_c$ over centroids $C$ as Eq. 5.14;
2 Sample $\hat{A}_c$ from a categorical distribution
    $\boldsymbol{z} = \textsc{one\_hot}\big(\operatorname{argmax}_i[g_i + \log\theta_i]\big), i \in \{1, \dots, k\}$,
    $g = -\log(-\log(u)), u \sim \textsc{uniform}(0, 1)$ ;
3 Differentially approximate $\hat{A}_c$ as Eq. 5.11;
4 Compute $\hat{K} = \hat{A}_c C^\top$ as Eq. 5.15;
5 Model stochastic attention $\hat{A}_v$ over value $V$ as Eq. 5.16;
6 Sample and approximate $\hat{A}_v$, similar to line 2 to 3;
7 Compute $H = \hat{A}_v V$ as Eq. 5.17;
```

### Why perform clustering on key heads?

Eq. 5.14 performs clustering on the key attention heads and outputs an attention distribution, and Eq. 5.15 tries to form a new head based on attention distribution and learned centroids. The goal is to make the original key heads to be stochastic, allowing attention distribution to have randomness for uncertainty estimation. This goal can also be accompanied by applying Eqs. 5.14 and 5.15 to the query while keeping the key unchanged. In that case, $\hat{a}_c$ can still be sampled stochastically based on query and centroids.

### Stochastic attention approximation

Eqs. 5.14 and 5.15 group the key heads into a fixed number of centroids and are reweighed by the mixture of centroids. As in [219], we can analyze the attention approximation error and derive that the key head attention difference is bounded.

**Proposition 5.4.1.** *Given two keys $k_i$ and $k_j$ such that $||k_i - k_j||_2 \leq \varepsilon$, stochastic key attention difference is bounded: $||\mathcal{G}(\tau^{-1}k_iC)) - \mathcal{G}(\tau^{-1}k_jC))||_2 \leq \tau^{-1}\varepsilon||C||_2$, where $\mathcal{G}$ is the Gumbel-Softmax function, and $||C||_2$ is the spectral norm of centroids. $\varepsilon$ and $\tau$ are constant numbers.*

**Proof** Same to the softmax function, which has a Lipschitz constant less than 1 [55], we have the following derivation:

$$\begin{aligned}
&||\mathcal{G}(\tau^{-1}k_iC)) - \mathcal{G}(\tau^{-1}k_jC))||_2 \\
&\leq ||\tau^{-1}k_iC - \tau^{-1}k_jC||_2 \\
&\leq \tau^{-1}\varepsilon||C||_2.
\end{aligned} \tag{5.18}$$

Proposition 5.4.1 shows that the $i$-th key assigned to $j$-th centroid can be bounded by its distance from $j$-th centroid. The keys that are close in Euclidean space have similar attention distribution over centroids.

## 5.5 Experimental setup

### 5.5.1 Research questions

We seek to answer the following questions based on experiments on three benchmark classification tasks:

(**RQ5.1**) Can the hierarchical stochastic transformers provide uncertain estimation while remaining predictive performance?

(**RQ5.2**) Can the hierarchical stochastic transformers outperforms the compared methods?

(**RQ5.3**) How well do the hierarchical stochastic transformers perform on ID and OOD datasets?

(**RQ5.4**) How well do the hierarchical stochastic transformers perform on trade-offs between predictive performance and uncertainty estimation?

### 5.5.2 Datasets

We verify the effectiveness of the proposed models on three NLP tasks: (i) sentiment analysis; (ii) linguistic acceptability; and (iii) slot filling.

We use the IMDB dataset[1] [130] for the sentiment analysis task. The standard IMDB has 25,000/25,000 reviews for training and testing, covering 72,062 unique words. For hyperparameter selection, we take 10% of training data as a validation set, leading to 22,500/2,500/25,000 data samples for training, validation, and testing. Besides, we use the customer review (CR) dataset [72], which has 500 samples to evaluate the proposed model in OOD settings.

We conduct a second group of experiments on a linguistic acceptability task with CoLA dataset[2] [227]. It consists of 8,551 training and 527 validation in-domain samples. As the labels of the test set arena are not publicly available, we split the 9,078 in-domain samples into train/validation/test with 7:1:2 randomly. Additionally, we use the provided 516 out-of-domain samples for uncertainty estimation.

We carry out a third group of experiments on a slot filling task with two benchmark datasets, i.e., ATIS [69] and SNIPS [29]. We follow the data division of [22, 61] for both datasets. ATIS records audio conversations of booking flights, which covers 120 slot labels across 21 types of intents. It is divided into 4,478, 500, and 893 utterances for training, validation, and testing, respectively. SNIPS embodies conversations from the personal voice assistant, which contains 72 slot labels across 7 types of intents. It is divided into 13,084, 700, and 700 utterances for training, validation, and testing, respectively. To assess our models for uncertainty estimation in both ID and OOD settings, we only use utterances with each type of intent for the OOD test set meanwhile leaving out the utterances with the same intent for training, validation, and ID test sets. We ignore the intents if their ID or OOD test sets are too small ($<$ 10 conversations). For ATIS,

[1] https://ai.stanford.edu/~amaas/data/sentiment/

[2] https://nyu-mll.github.io/CoLA/

we study 8 selected intents, namely Abbreviation, Airfare, Airline, Airport, Capacity, Distance, Flight, and GroundService. For SNIPS, we study all 7 intents, i.e., AddToPlaylist, BookRestaurant, GetWeather, PlayMusic, RateBook, SearchCreativeWork, and SearchScreeningEvent.

### 5.5.3 Evaluation metrics and objectives

We report different commonly used evaluation metrics for different tasks:

- *Accuracy* is used in sentiment analysis task, which is defined as the number of correct predictions divided by the total number of predictions.
- *Matthews correlation coefficient* (MCC) [137] (a.k.a. phi coefficient) is used in linguistic acceptability task, which is measures the difference between the predicted values and actual values.
- *F1*-score is used in slot filling task, which is the harmonic mean of the precision and recall of the test set.

To evaluate the effectiveness of the proposed models, we summarize the following objectives:

- To evaluate the predictive performance of models on in-domain datasets. High predictive scores and low uncertainty scores are desired.
- To compare the model generalization from in-domain to out-of-domain datasets. High scores are desired.
- To estimate the uncertainty of the models on out-of-domain datasets. High uncertainty scores are desired.
- To measure the model capability in learning the predictive performance and uncertainty estimation trade-off.

### 5.5.4 Compared methods

We compare the following methods in our experimental setup:

- TRANS [216] is the vanilla transformer with deterministic self-attention.
- MC-DROPOUT [54] uses dropout [204] as a regularizer to measure the prediction uncertainty.
- ENSEMBLE [99] averages over multiple independently trained transformers.
- STO-TRANS is the proposed method in this work that the attention distribution over values is stochastic;
- H-STO-TRAN is the proposed method in this work uses hierarchical stochastic self-attention, i.e., the stochastic attention from key heads to a learnable set of centroids and the stochastic attention to value, respectively.

### 5.5.5 Implementation details

We implement all models in this work by PyTorch [163]. The models are optimized with Adam [91]. For each trained model, we sample 10 predictions (i.e., run the inference 10 times); the mean and variance (or standard deviation) of results are reported. The uncertainty information is quantified with variance (or standard deviation).

For sentiment analysis, we use 1 layer with 8 heads; both the embedding size and the hidden dimension size are 128. We train the model with a learning rate of 1e-3, a batch size of 128, and a dropout rate of 0.5/0.1. We evaluate the models at each epoch, and the models are trained with a maximum of 50 epochs.

For linguistic acceptability, we use 8 layers and 8 heads, the embedding size is 128, and the hidden dimension is 512. We train the model with a learning rate of 5e-5, a batch size of 32, and a dropout rate of 0.1. We train the models with a maximum of 2,000 epochs and evaluate the models at every 50 epochs. The model selection is performed based on the validation dataset according to predictive performance.

For slot filling, we use 12 layers and 12 heads; both the embedding size and the hidden size are 768. We train the model with a learning rate of 5e-5, a batch size of 32, and a dropout rate of 0.1. We evaluate models at each epoch and train the model with a maximum epoch of 400. The code is available online.[3]

## 5.6 Results

### 5.6.1 Results on the sentiment analysis task

Table 5.1 presents the predictive performance and uncertainty estimation on IMDB (in-domain, ID) and CR (out-of-domain, OOD) dataset, evaluated by accuracy.

First, STO-TRANS and H-STO-TRANS are able to provide uncertainty information, as well as maintain and even slightly outperform the predictive performance of TRANS. Specially, STO-TRANS ($\tau = 40$) and H-STO-TRANS ($\tau_1 = 1$, $\tau_2 = 30$) outperforms TRANS ($\eta = 0.1$) by 0.42% and 0.66% on the ID dataset. In addition, they allow us to measure uncertainty via predictive variances. This is because they inject randomness directly into self-attentions. However, TRANS has no access to uncertainty information due to its deterministic nature.

Second, STO-TRANS is struggling to learn a good trade-off between ID predictive performance and OOD uncertainty estimation performance. With a small temperature $\tau = 1$, STO-TRANS gives good uncertainty information, but we observe that the ID predictive performance drops. When $\tau$ approaches $\sqrt{d/h}$ (the original scaling factor in the vanilla transformer), STO-TRANS achieves better performance on the ID dataset, but lower performance on the OOD dataset. We conjecture that the randomness in STO-TRANS is solely based on the attention distribution over values, and this is not enough for learning the trade-off.

Third, H-STO-TRANS achieves a better accuracy-uncertainty trade-off compared with STO-TRANS. For instance, with $\tau_1 = 1, \tau_2 = 20$, H-STO-TRANS achieves 87.63% and 67.14%, which outperform the corresponding numbers of STO-TRANS for both the

[3] `https://github.com/amzn/sto-transformer`

Table 5.1: The predictive performance and uncertainty estimation of models on IMDB (ID) and CR (OOD) dataset. The uncertainty estimation is performed by running forward pass inference by 10 runs; then, the uncertainty is quantified by the standard deviation across runs. For the ensemble, the results are averaged over 10 models that are independently trained with random seeds. Dropout is used in the inference of MC-DROPOUT, and $\eta$ is the dropout rate. For the remaining methods, dropout is not used in inference. The $\triangledown$ID (%) and $\triangledown$OOD (%) present the predictive performance difference to TRANS ($\eta = 0.1$).

| | **ID (%)** | **OOD (%)** | $\triangledown$ID (%) | $\triangledown$OOD (%) |
|---|---|---|---|---|
| TRANS ($\eta = 0.1$) | 87.00 | 65.00 | / | / |
| TRANS ($\eta = 0.5$) | 87.51 | 63.40 | 0.51 ↑ | 1.60 ↓ |
| MC-DROPOUT ($\eta = 0.5$) | $86.06 \pm 0.087$ | $63.38 \pm 1.738$ | 0.94 ↑ | 1.62 ↓ |
| MC-DROPOUT ($\eta = 0.1$) | $87.01 \pm 0.075$ | $63.38 \pm 0.761$ | 0.10 ↑ | 1.62 ↓ |
| ENSEMBLE | $86.89 \pm 0.230$ | $64.20 \pm 1.585$ | 0.11 ↓ | 0.80 ↓ |
| STO-TRANS ($\tau = 1$) | $82.62 \pm 0.092$ | $67.92 \pm 0.634$ | 4.38 ↓ | 2.92 ↑ |
| STO-TRANS ($\tau = 40$) | $87.42 \pm 0.022$ | $63.78 \pm 0.289$ | 0.42 ↑ | 1.22 ↓ |
| H-STO-TRANS ($\tau_1 = 1, \tau_2 = 20$) | $87.63 \pm 0.017$ | $67.14 \pm 0.400$ | 0.63 ↑ | 2.14 ↑ |
| H-STO-TRANS ($\tau_1 = 1, \tau_2 = 30$) | $87.66 \pm 0.022$ | $66.72 \pm 0.271$ | 0.66 ↑ | 1.72 ↑ |

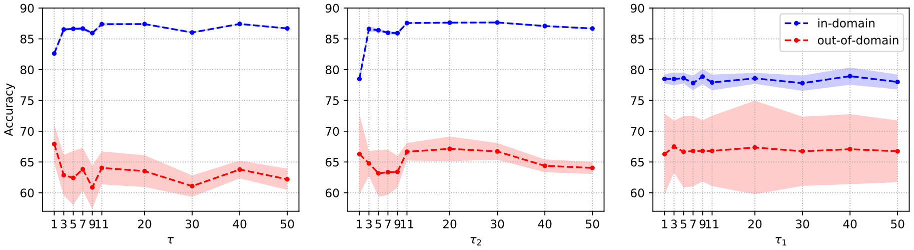


Figure 5.3: Experiments with the hyperparameter $\tau$. Left: STO-TRANS with different $\tau$. The randomness is solely based on the sampling of attention distribution over values. While uncertainty information is captured, STO-TRANS has difficulties in learning the trade-off between in-domain and out-of-domain performance. Middle: The hyperparameter tuning of $\tau_1$ and $\tau_2$ in H-STO-TRANS. $\tau_1$ controls the concentration on centroids and $\tau_2$ controls the concentration on values.

ID and OOD datasets. It also outperforms MC-DROPOUT and ENSEMBLE; Specifically, H-STO-TRANS improves by 0.62%-1.6% and 2.52%-3.76% on the ID and OOD datasets, respectively. On the OOD dataset, while MC-DROPOUT and ENSEMBLE exhibit higher uncertainty (measured by standard deviation) across runs, the accuracy is lower than that of TRANS ($\eta = 0.1$), STO-TRANS ($\tau = 1$) and H-STO-TRANS. It is due to a better way of learning two types of randomness: one from sampling over a set of learnable centroids and the other from sampling attention over values.

Figure 5.3 reports the hyperparameter tuning of $\tau_1$ and $\tau_2$. The goal is to find a reasonable combination to achieve high predictive performance on both ID and OOD

Table 5.2: Performance of compared models on the CoLA dataset. We set all temperature values $\tau_1 = 1$ and $\tau_2 = 1$. The $\bigtriangledown$ID (%) and $\bigtriangledown$OOD (%) present the predictive performance and difference to TRANS ($\eta = 0.1$), respectively.

| **Model** | **ID (%)** | **OOD (%)** | **$\bigtriangledown$ID (%)** | **$\bigtriangledown$OOD (%)** |
|---|---|---|---|---|
| TRANS ($\eta = 0.1$) | 20.09 | 16.46 | / | / |
| MC-DROPOUT ($\eta = 0.1$) | $19.91 \pm 0.40$ | $16.70 \pm 2.21$ | 0.18 ↓ | 0.24 ↑ |
| MC-DROPOUT ($\eta = 0.05$) | $20.03 \pm 0.30$ | $17.11 \pm 1.21$ | 0.06 ↓ | 0.65 ↑ |
| ENSEMBLE | $21.20 \pm 2.59$ | $16.73 \pm 4.92$ | 1.11 ↑ | 0.27 ↑ |
| STO-TRANS | $23.27 \pm 0.75$ | $15.25 \pm 4.65$ | 3.18 ↑ | 1.21 ↓ |
| H-STO-TRANS | $20.52 \pm 0.76$ | $16.49 \pm 4.08$ | 0.43 ↑ | 0.03 ↑ |

datasets. To simplify the tuning work, we fix the $\tau_1 = 1$ and then change $\tau_2$ with different values, and vice versa. As we can see, the combination of a small $\tau_1$ and a large $\tau_2$ performs better than the other way around. We think this is because $\tau_2$ is in the latter stage and has bigger effects on the predictive performance. However, if we remove $\tau_1$, H-STO-TRANS reverts to STO-TRANS, where the accuracy-uncertainty trade-off is not well learned, as shown in Figure 5.3 (Left).

### 5.6.2 Results on the linguistic acceptability task

Table 5.2 shows the performance of compared models on both the in-domain (ID) and out-of-domain (OOD) sets of the CoLA dataset, evaluated by MCC.

First, STO-TRANS and H-STO-TRANS obtain comparable performance as well as provide uncertainty information, compared with TRANS. To be specific, STO-TRANS and H-STO-TRANS improves 3.18% and 0.43% of MCC on the ID dataset compared with deterministic TRANS, respectively.

Second, STO-TRANS achieves the best performance on the ID dataset but the worst performance on the OOD dataset. Although STO-TRANS outperforms TRANS, the best MC-DROPOUT, ENSEMBLE by 3.18%, 3.24%, 2.07% of MCC on the ID dataset, its performance drops by 1.21%, 1.86%, and 1.48%, correspondingly on the OOD dataset. This further verifies our conjecture that the randomness is only introduced to attention distribution over values and is insufficient for learning the trade-off of ID and OOD data.

Third, H-STO-TRANS is able to learn a better trade-off between prediction and uncertainty. More precisely, the performance improves 0.43% and 0.03% of MCC on the ID and OOD datasets, respectively. H-STO-TRANS is 0.49% better than MC-DROPOUT ($\eta = 0.05$), but 0.68% worse than ENSEMBLE on the ID dataset. Given that ENSEMBLE shows high uncertainty on the ID dataset and MC-DROPOUT ($\eta = 0.05$) has low uncertainty on the OOD dataset, this is not desired. Therefore, H-STO-TRANS strikes a better balance across the objectives. In the context of this task, this means high MCC, low variance on the ID dataset, and high MCC, high variance on the OOD dataset.

Table 5.3 gives some predictions of test samples with H-STO-TRANS. What we observe are two aspects: (i) In general, ID predictions have lower variances in terms

Table 5.3: Illustration of predictions with H-STO-TRANS. The predictions for the ID (top) and OOD (bottom) samples are measured by the probability of being correct for each prediction and the number of correct predictions.

| **Examples (Labels)** | **Prob. Corr.** | **Corr./Total** |
|---|---|---|
| no man has ever beaten the centaur. (1) | 0.75 ± 0.001 | 10/10 |
| nora sent the book to london (1) | 0.65 ± 0.007 | 10/10 |
| sally suspected joe, but he did n't holly. (1) | 0.60 ± 0.008 | 8/10 |
| kim is eager to recommend. (0) | 0.41 ± 0.011 | 3/10 |
| he analysis her was flawed (0) | 0.24 ± 0.003 | 0/10 |
| sandy had read how many papers ? ! (1) | 0.67 ± 0.010 | 10/10 |
| which book did each author recommend ? (1) | 0.58 ± 0.010 | 7/10 |
| she talked to harry , but i do n't know who else . (1) | 0.52 ± 0.013 | 4/10 |
| john is tall on several occasions . (0) | 0.42 ± 0.005 | 1/10 |
| they noticed the painting , but i do n't know for how long . (0) | 0.28 ± 0.003 | 0/10 |

of the probability of being correct. For "10/10" (10 correct predictions out of 10 total predictions) prediction cases, the ID examples have a higher probability score than the ones in OOD data. Also, we find there are much less number of "10/10" prediction cases in the OOD dataset than that in the ID dataset. (ii) For the ID dataset, either with high or low probability scores, we can see low variances; we see more "10/10" (tend to be confidently correct) or "0/10" (tend to be confidently incorrect) cases. As expected, for both cases, the variance is relatively low compared to the probability of around 0.5. In deterministic models, we are not able to access this kind of information which would imply how confident are the transformer models toward predictions.

### 5.6.3 Results on the slot filling task

Figure 5.4 depicts the performance of the compared models on the ATIS subsets, grouped by selected intents for both in-domain (ID) and out-of-domain (OOD) settings, evaluated by F1 scores.

First, STO-TRANS and H-STO-TRANS enable us to estimate uncertainty scores, as well as maintain and even outperform the predictive performance of TRANS. Intuitively, STO-TRANS greatly outperforms others for all intents in ID settings, and it is quite competitive in most OOD settings (except for Capacity and Airport) and always shows H-shaped bars. H-STO-TRANS is basically on par with or even slightly beats TRANS always with extra H-shaped bars.

Second, STO-TRANS hardly makes a good trade-off in performance between ID and OOD data. For example, for Capacity and Airport, STO-TRANS achieves the highest predictive performance on ID data, but it drops to worst on OOD data. In this case, H-STO-TRANS performs better predictive performance as well as uncertainty scores.

Third, H-STO-TRANS tends to provide better uncertainty estimations with longer H-shaped bars in OOD data. However, the overall gains of H-STO-TRANS are inferior to STO-TRANS for slot filling. We blame this on the increase of task difficulty incurred by as many as 120 predictive labels, while the previous two tasks only have 2 predictive

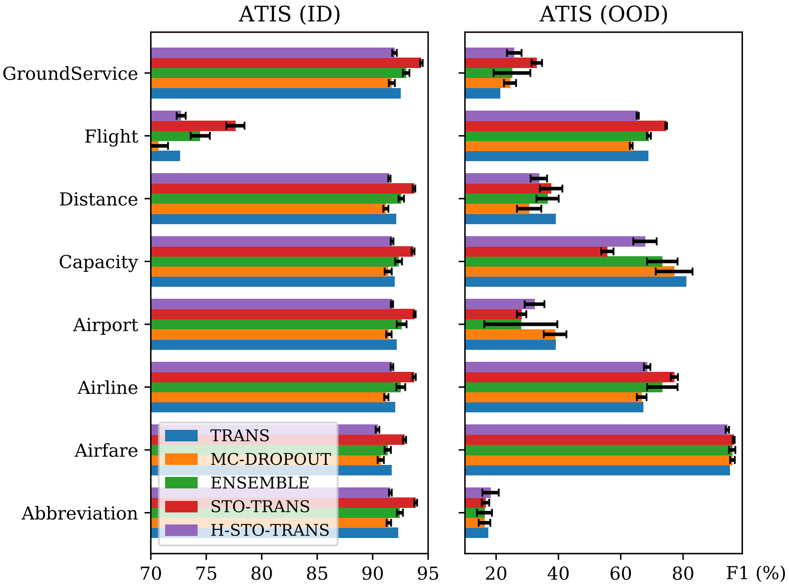


Figure 5.4: Performance of compared models on the ATIS subsets grouped by 8 selected intents. The x-axis indicates the F1 score, and the y-axis represents intent types. The length of the bars denotes the predictive performance, and the length of the H-shaped bars indicates the uncertainty scores.

labels.

Fourth, besides uncertain estimation performance, ENSEMBLE has also shown competitive predictive performance on both ID and OOD data. MC-DROPOUT slightly inferior to STO-TRANS for predictive performance, but provides strong performance of uncertain estimation, especially in OOD setting.

Figure 5.5 shows the performance of the compared models on SNIPS subsets, divided by all intents for both in-domain (ID) and out-of-domain (OOD) settings, evaluated by F1 scores.

First, STO-TRANS is able to estimate uncertainty and achieves the highest predictive performance among the compared models for both ID and OOD data. The only exception is SearchCreativeWork. This indicates the performance of STO-TRANS is not promising if its base model, TRANS has a poor performance.

Second, the predictive performance of H-STO-TRANS mostly drops compared with other models in both ID and OOD settings. Similar to the situation on the ATIS dataset, task difficulty increases with more predictive labels.

Third, ENSEMBLE obtains the second best competitive predictive performance, as well as uncertain estimation performance, on both ID and OOD data. MC-DROPOUT is less effective than STO-TRANS for predictive performance, but provides promising performance of uncertain estimation, especially in the OOD setting.

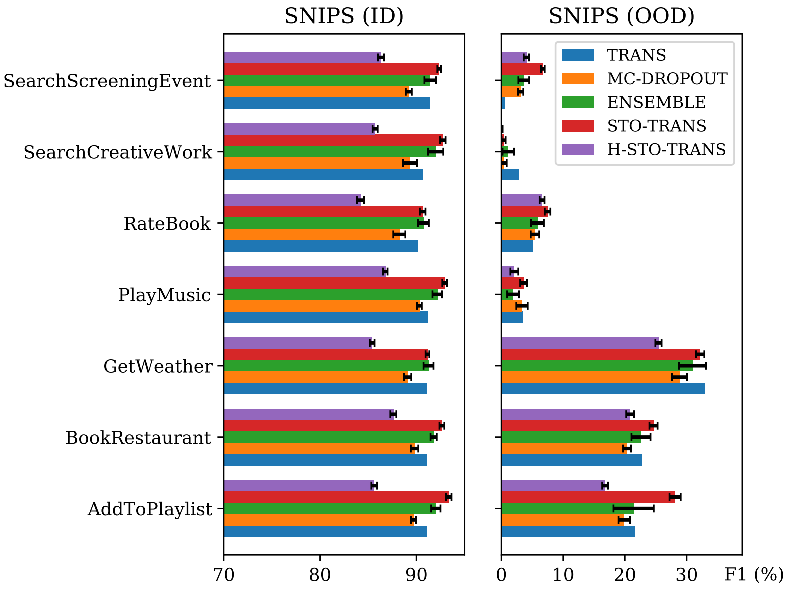


Figure 5.5: Performance of compared models on SNIPS subsets grouped by all 7 different intents. The x-axis indicates the F1 score, and the y-axis represents intent types. The length of the bars denotes the predictive performance, and the length of the H-shape bars indicates the uncertainty scores.

## 5.7 Discussion

In this section, we take a step back and discuss the results from the point of view of comparable models and tasks.

From a model perspective, most transformer variants are still deterministic, while many extension of transformers have recently been proposed (e.g., [37, 66, 77, 94, 102]). Our goal in this chapter has been to equip transformers in a stochastic way to estimate uncertainty while retaining the original predictive performance. This requires a special design in order to achieve the two goals without adding a major computational overhead to model training and inference like Ensemble or Bayesian neural networks (BNNs). The complexity gain of our method to its deterministic version is modest and requires an additional matrix $C \in \mathbb{R}^{d_h \times c}$. This is more efficient than Ensemble and BNNs, which gives N ($N \geq 2$ for Ensemble and $N = 2$ for BNNs) times more weights.

From a task perspective, sentiment analysis and linguistic acceptability are binary classification tasks, while slot filling is a multi-class classification task. All tasks use the proposed stochastic transformers as an encoder, which is able to capture useful representation for classification while injecting stochasticity. The main difference is that slot filling has a larger solution space than sentiment analysis and linguistic acceptability, which leads to an increase in task difficulty.

## 5.8 Conclusion

In this chapter, we have introduced a novel, simple yet effective way to enable transformers with uncertainty estimation as an alternative to MC dropout and ensembles. We have proposed variants of transformers based on two stochastic self-attention mechanisms: (i) injecting stochasticity into the stochastic attention over values; and (ii) forcing key heads to pay stochastic attention to a set of learnable centroids.

Our experimental results show that the proposed approach learns good trade-offs between in-domain predictive performance and out-of-domain uncertainty estimation performance on three NLP benchmark tasks, and it outperforms baselines. Together, these results provide an answer to the leading research question for this chapter and show that we can enable collaborative agents with the capability of uncertainty estimation towards trustworthy systems, which is an important step for their future adoption.

As to broader implications of the work in this chapter, the proposed stochastic transformers can easily be used for many NLP tasks, especially where vanilla transformers have been shown to be effective. The uncertain estimation scores are meaningful and useful for final decision-making by people, and they also can be used as additional input to other models for trustable systems.

An important limitation of this work is that we have only assessed the proposed models as an encoder. Hence, for future work, it is important to evaluate the proposed models on other transformer architecture (e.g., a transformer with an encoder-decoder) tasks (e.g., dialogue response generation).

Next, we conclude the thesis.

# 6
# Conclusions

In this chapter, we reflect on the research questions we formulated in Section 1.1 and summarize the main findings based on the research chapters in Section 6.1. Then, in Section 6.2, we propose future research directions that build on the work in the thesis.

## 6.1 Main Findings

**RQ1** Can multiple dialogue agents collaborate effectively to improve the performance of a single-module agent?

As an answer to **RQ1**, we have proposed a mixture-of-generators network (MoGNet) for dialogue response generation (DRG). We assume that multiple expert agents are specialized generators for diverse intents, and that a chair agent decides each final token of a response by a mixture of experts with two collaboration mechanisms, i.e., retrospective mixture-of-generators (RMoG) and a prospective mixture-of-generators (PMoG). To effectively train mixture-of-generators network (MoGNet), we have devised a global-and-local (GL) learning scheme that forces each expert to minimize a local loss for specialization and that makes the chair collaborate with all experts to optimize the global loss for generalization. We have conducted extensive experiments, analyses, as well as automatic and human evaluation as part of an empirical study on the MultiWOZ benchmark dataset. Our main findings are as follows:

(1) MoGNet, which is composed of collaborative models, can significantly outperform single-module agents when generating a dialogue response with a collaboration of expert models.

(2) How to group and integrate multiple dialogue agents are key factors that determine the gain of MoGNet.

(3) The GL learning scheme enables local expert losses for specialization and the global chair loss for generalization, so that it greatly influences the effectiveness of MoGNet, in terms of model training.

**RQ2** Can multiple users collaborate successfully to improve the quality of a dialogue for each single user?

As an answer to **RQ2**, we have proposed a cooperative memory network (CoMemNN) to gradually and simultaneously address user profile enrichment (UPE) and improve personalized dialogue response selection (DRS). The UPE module enriches incomplete user profiles by using collaborative information from neighboring users in addition to ongoing dialogues. The DRS module uses the enriched user profiles to simultaneously improve the quality of personalized responses. We have conducted extensive experiments and analyses on the personalized bAbI dialogue datasets and simulated datasets with various degrees of incompleteness. The main findings are as follows:

(1) CoMemNN, building on information from collaborative users, can gradually enrich user profiles as dialogues progress and simultaneously improve the quality of personalized dialogues based on the enriched profiles.

(2) The robustness of CoMemNN can be attributed to the fact that the UPE module can effectively enrich user profiles in the presence of incomplete user profiles.

(3) A multiple-hop learning mechanism can enhance the training of CoMemNN model.

**RQ3** Can multiple languages be used in a collaborative way to improve the performance of each single language?

To answer **RQ3**, we have proposed a mixture-of-languages routing (MOLR) paradigm under a collaborative chair-experts framework. Each expert agent can be either monolingual or cross-lingual, and a chair agent conducts a mixture of experts for globally optimizing multilingual expert agents. The paradigm contains four functional components, i.e., input embeddings, language model, pairwise alignment, and mixture-of-languages. First, we use mT5 [242] as the backbone of our base model for the former two components. We conduct pairwise alignment to exploit relationships between every two language routes and bridge the language gap. Next, we globally optimize a mixture of language routing with two collaboration policies, i.e., route-addressing and parameter-sharing. After that, we quantify language characteristics of unity and diversity by similarity metrics, i.e., genetic similarity and word and sentence similarity based on embeddings. We have conducted extensive experiments and analyses on two benchmark datasets, i.e., the multilingual DST dataset [148] and the NLU dataset [190]. Our main findings are as follows:

(1) MOLR, exploiting a collaboration between languages, can globally and simultaneously optimize the multilingual TDS performance.

(2) Gains of MOLR mainly come from multilingual data argumentation, the modeling of language characteristics, and mixture-of-language routing.

(3) Gradually crossing the language chasm is better: a smaller gap (or a higher degree of similarity) between the source language and the pivot language is usually beneficial for the overall performance.

**RQ4** Can we enable collaborative agents with the capability of uncertainty estimation towards trustworthy systems?

To answer **RQ4**, we have proposed two stochastic transformers: (i) STO-TRANS, which enables each head to perform stochastic attention over values using the Gumbel-Softmax trick; (ii) H-STO-TRANS, as an extension of STO-TRANS, which forces each head to pay stochastic attention to a set of learnable centroids, and each centroid performs stochastic attention over values. We have conducted extensive experiments and analyses on three benchmark tasks, i.e., sentiment analysis (SA), linguistic acceptability (LA), and slot filling (SF). The main findings are as follows:

(1) Both STO-TRANS and H-STO-TRANS enable vanilla transformers with stochastic self-attention to provide uncertainty information while maintaining predictive performance.

(2) A theoretical proof has shown that the proposed self-attention approximation is upper bounded, and the key attention heads that are close in terms of Euclidean distance have a similar attention distribution over centroids.

(3) The additional benefit of H-STO-TRANS compared with STO-TRANS is the capability to trade off the performance between in-domain (ID) prediction and out-of-domain (OOD) uncertainty estimation, evaluated on the aforementioned three benchmark tasks.

## 6.2 Future work

In this section, we identify a number of underexplored topics in the area of collaborative task-oriented dialogue systems (CTDSs) by reviewing the CTDS models we have proposed.

### 6.2.1 Self-supervised partition and collaboration

MoGNet outperforms other methods in conducting complex dialogues with multiple complex intents. However, the pre-defined partition of intents affects the final results a lot, based on experimental results. For example, MoGNet partitioned by “domains” greatly outperforms MoGNet partitioned by “system actions.” This might be caused by the different levels of granularity of the different pre-defined partitions. It would be interesting to explore how to automatically learn the collaborative agents and adopt these agents to task-oriented dialogue systems (TDSs). We have found that collaboration policies of the dialogue agents are important for the final performance in MoGNet, CoMemNN, and MOLR. So it is also valuable to study the effectiveness of different collaborative policies in the automatic partition settings. Overall, we expect to improve collaborative TDSs by automatically learning partitions and collaboration in a self-supervised manner.

To sum up, interesting research questions along this future direction include but are not limited to: (i) How to effectively learn the partitions of intents? (ii) How to group partitions of intents by appropriate granularity? (iii) How to devise effective collaborative policies given a partition of intents? (iv) How well do TDSs working with automatically inferred intent partitions perform compared with the pre-defined collaborative TDSs?

### 6.2.2 Uncertainty estimation in collaborative agents

Stochastic transformers are able to equip vanilla transformers with uncertainty estimation capabilities while maintaining predictive performance. We regard a vanilla transformer as a sequential stack of collaborative agents, where each agent is a transformer block. It is still unexplored how to estimate uncertainty for trustworthy systems in other collaborative agent models, e.g., MoGNet, CoMemNN and MOLR. Intuitively, the overall uncertainty of a collaborative TDS is obtained from all collaborative agents and is influenced by different collaboration policies. This increases the need for uncertainty estimation for trustworthy systems. We have found that partition aspects and collaboration policies are key factors for the predictive performance of collaborative agents. So partition aspects and collaboration policies might also influence the uncertainty estimation. Overall, we believe it is important to equip collaborative TDSs with uncertainty estimation capabilities while retaining the predictive performance compared with single-module agents.

To sum up, interesting research questions along this future direction include but are not limited to: (i) How to estimate uncertainty and evaluate the estimation performance for dialogue models whose constituents are collaborative agents? (ii) How do different partitions aspects influence the uncertainty estimation of collaborative agents? (iii) How do different collaboration policies influence the uncertainty estimation of collaborative agents? (iv) How well do collaborative agents perform on uncertainty estimation compared with a single-module agent?

### 6.2.3 Collaboration effectiveness and efficiency

In a collaborative setting, adding more agents may increase concerns about efficiency compared to a single-module agent. We have explored several lightweight expert agents, i.e., RNNs in MoGNet, memory networks in CoMemNN, and the small version of mT5 in bilingual MOLR, transformer blocks in stochastic transformers. The collaborative models achieve good effectiveness; at the same time, they maintain acceptable efficiency. However, we have found that if we swap mT5 with the base or large versions, the computational requirements increase dramatically, e.g., GPU memory, computational time, and computational costs. One potential solution is to train lots of lightweight agents in a parallel way to make full use of cheap equipment. So far, the key factors that influence efficiency are unexplored. We call for future work to explore more appropriate solutions to trade off between collaboration effectiveness and efficiency. Last but not least, while large pre-trained language models are being used widely for many NLP tasks, it is valuable to investigate when to choose small collaborative agents rather than a large single-module agent, e.g., BERT, GPT2, mT5, etc.

To sum up, interesting research questions along this future direction include but are not limited to: (i) Can dialogue models with collaborative agents improve efficiency while maintaining effectiveness at the same time? (ii) What are the key factors that can influence the efficiency of collaborative dialogue agents? (iii) How to choose between small collaborative agents and a large single-module agent?

# Bibliography

[1] K. Ahmed and L. Torresani. Star-caps: Capsule networks with straight-through attentive routing. In *NeurIPS*, volume 32, pages 9101–9110, 2019. (Cited on page 79.)

[2] E. H. Almansor and F. K. Hussain. Survey on intelligent chatbots: State-of-the-art and future research directions. In *CISIS*, pages 534–543. Springer, 2019. (Cited on page 1.)

[3] N. Arivazhagan, A. Bapna, O. Firat, D. Lepikhin, M. Johnson, M. Krikun, M. X. Chen, Y. Cao, G. Foster, C. Cherry, et al. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*, 2019. (Cited on page 5.)

[4] A. Ashukha, A. Lyzhov, D. Molchanov, and D. Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *ICLR*, 2019. (Cited on page 81.)

[5] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. (Cited on page 17.)

[6] S. R. Baker, N. Bloom, S. J. Davis, and S. J. Terry. Covid-induced economic uncertainty. Technical report, National Bureau of Economic Research, 2020. (Cited on page 77.)

[7] A. Bapna, G. Tur, D. Hakkani-Tur, and L. Heck. Sequential dialogue context modeling for spoken language understanding. In *SIGDIAL*, pages 103–114, 2017. (Cited on page 13.)

[8] L. Beinborn and R. Choenni. Semantic drift in multilingual representations. *Computational Linguistics*, 46(3):571–603, 2020. (Cited on page 55.)

[9] L. Beinborn, T. Zesch, and I. Gurevych. Cognate production using character-based machine translation. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 883–891, 2013. (Cited on page 55.)

[10] J. Bjerva, R. Östling, M. H. Veiga, J. Tiedemann, and I. Augenstein. What do language representations really represent? *Computational Linguistics*, 45(2):381–389, 2019. (Cited on page 55.)

[11] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *ICML*, pages 1613–1622. PMLR, 2015. (Cited on pages 78 and 80.)

[12] A. Bordes and J. Weston. Learning end-to-end goal-oriented dialog. In *ICLR*, 2017. (Cited on pages 1, 29, and 39.)

[13] P. Budzianowski, I. Casanueva, B.-H. Tseng, and M. Gasic. Towards end-to-end multi-domain dialogue modelling. Technical report, Cambridge University, 2018. (Cited on pages 11, 16, 20, and 21.)

[14] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gasic. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*, pages 5016–5026, 2018. (Cited on pages 4, 11, 12, 15, 19, 20, and 21.)

[15] S. Burger, K. Weilhammer, F. Schiel, and H. G. Tillmann. Verbmobil data collection and annotation. In *Verbmobil: Foundations of speech-to-speech translation*, pages 537–549. Springer, 2000. (Cited on page 50.)

[16] H. C. Carneiro, F. M. França, and P. M. Lima. Multilingual part-of-speech tagging with weightless neural networks. *Neural Networks*, 66:11–21, 2015. (Cited on page 55.)

[17] CAsT. Trec conversational assistance track. `https://www.treccast.ai`, 2022. (Cited on page 1.)

[18] D. Chandrasekaran and V. Mago. Evolution of semantic similarity—a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–37, 2021. (Cited on pages 55 and 61.)

[19] G.-L. Chao and I. Lane. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. *Proceedings of Interspeech*, 2019. (Cited on pages 2, 51, and 64.)

[20] H. Chen, X. Liu, D. Yin, and J. Tang. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter*, 19(2):25–35, 2017. (Cited on pages 1, 2, 14, 20, 29, 49, and 55.)

[21] P.-C. Chen, T.-C. Chi, S.-Y. Su, and Y.-N. Chen. Dynamic time-aware attention to speaker roles and contexts for spoken language understanding. In *ASRU Workshop*, pages 554–560, 2017. (Cited on pages 13 and 14.)

[22] Q. Chen, Z. Zhuo, and W. Wang. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*, 2019. (Cited on page 86.)

[23] W. Chen, J. Chen, Y. Su, X. Wang, D. Yu, X. Yan, and W. Y. Wang. Xl-nbt: A cross-lingual neural belief tracking framework. In *EMNLP*, pages 414–424, 2018. (Cited on pages 56 and 75.)

[24] W. Chen, J. Chen, P. Qin, X. Yan, and W. Y. Wang. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *ACL*, pages 3696–3709, 2019. (Cited on page 15.)

[25] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014. (Cited on page 16.)

[26] A. Conneau and G. Lample. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32, 2019. (Cited on page 75.)

[27] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data. In *ICLR*, 2018. (Cited on page 75.)

[28] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. In *ACL*, pages 8440–8451, 2020. (Cited on pages 51, 54, and 75.)

[29] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, et al. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, 2018. (Cited on page 86.)

[30] P. Crook, A. Marin, V. Agarwal, K. Aggarwal, T. Anastasakos, R. Bikkula, D. Boies, A. Celikyilmaz, S. Chandramohan, Z. Feizollahi, R. Holenstein, M. Jeong, O. Z. Khan, Y.-B. Kim, E. Krawczyk, X. Liu, D. Panic, V. Radostev, N. Ramesh, J.-P. Robichaud, A. Rochette, S. L., and R. Sarikaya. Task completion platform: A self-serve multi-domain goal oriented dialogue platform. In *NAACL*, pages 47–51, 2016. (Cited on page 14.)

[31] D. Crystal. Two thousand million? *English today*, 24(1):3–6, 2008. (Cited on page 49.)

[32] R. Csaky and G. Recski. The gutenberg dialogue dataset. In *EACL*, pages 138–159, 2021. (Cited on page 50.)

[33] M. Cysouw. Predicting language-learning difficulty. In *Approaches to Measuring Linguistic Differences*. De Gruyter, 2013. (Cited on page 55.)

[34] R. Dabre, A. Imankulova, M. Kaneko, and A. Chakrabarty. Simultaneous multi-pivot neural machine translation. *arXiv preprint arXiv:2104.07410*, 2021. (Cited on page 50.)

[35] E. Dabrowska. What exactly is universal grammar, and has anyone seen it? *Frontiers in psychology*, 6: 852, 2015. (Cited on page 55.)

[36] M. Daniel. Linguistic typology and the study of language. In *The Oxford handbook of linguistic typology*. Oxford University Press, 2011. (Cited on page 55.)

[37] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser. Universal transformers. In *ICLR*, 2018. (Cited on page 93.)

[38] A. Der Kiureghian and O. Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2): 105–112, 2009. (Cited on page 81.)

[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. (Cited on pages 15, 51, 54, 56, 75, 76, and 77.)

[40] T. G. Dietterich. Ensemble methods in machine learning. In *MCS Workshop*, pages 1–15, 2000. (Cited on pages 3, 12, and 14.)

[41] B. Ding, J. Hu, L. Bing, M. Aljunied, S. Joty, L. Si, and C. Miao. Globalwoz: Globalizing multiwoz to develop multilingual task-oriented dialogue systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1639–1657, 2022. (Cited on pages 50, 54, 55, 56, and 57.)

[42] R. M. Dixon. *I am a linguist: with a foreword by Peter Matthews*. Brill, 2010. (Cited on page 54.)

[43] J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. Miller, A. Szlam, and J. Weston. Evaluating prerequisite qualities for learning end-to-end dialog systems. In *ICLR*, 2016. (Cited on page 35.)

[44] O. Dušek and F. Jurcıcek. A context-aware natural language generator for dialogue systems. In *SIGDIAL*, pages 185–190, 2016. (Cited on page 13.)

[45] L. El Asri, H. Schulz, S. K. Sarma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, and K. Suleman. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *SIGDIAL*, pages 207–219, 2017. (Cited on page 1.)

[46] M. Eric, L. Krishnan, F. Charette, and C. D. Manning. Key-value retrieval networks for task-oriented dialogue. In *SIGDIAL*, pages 37–49, 2017. (Cited on pages 1, 14, and 33.)

[47] Y. Fan, X. Luo, and P. Lin. A survey of response generation of dialogue systems. *International Journal of Computer and Information Engineering*, 14(12):461–472, 2020. (Cited on pages 2 and 4.)

[48] J. Ficler and Y. Goldberg. Controlling linguistic style aspects in neural language generation. In *Workshop on Stylistic Variation*, pages 94–104, 2017. (Cited on pages 29 and 31.)

[49] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks.

In *ICML*, pages 1126–1135. PMLR, 2017. (Cited on page 57.)

[50] W. T. Fitch. Unity and diversity in human language. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1563):376–388, 2011. (Cited on page 54.)

[51] A. Y. Foong, Y. Li, J. M. Hernández-Lobato, and R. E. Turner. 'in-between' uncertainty in bayesian neural networks. In *ICML Workshop*, 2019. (Cited on pages 79 and 80.)

[52] A. François. Trees, waves and linkages: Models of language diversification. In *The Routledge handbook of historical linguistics*, pages 161–189. Routledge, 2015. (Cited on page 54.)

[53] P. Fung and T. Schultz. Multilingual spoken language processing. *IEEE Signal Processing Magazine*, 25(3):89–97, 2008. (Cited on pages 5, 49, and 50.)

[54] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, pages 1050–1059. PMLR, 2016. (Cited on pages 79, 80, 84, and 87.)

[55] B. Gao and L. Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017. (Cited on page 85.)

[56] C. Gao, W. Lei, X. He, M. de Rijke, and T.-S. Chua. Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2:100–126, 2021. (Cited on page 1.)

[57] J. Gao, C. Xiong, and P. Bennett. Recent advances in conversational information retrieval. In *SIGIR*, pages 2421–2424, 2020. (Cited on page 1.)

[58] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021. (Cited on page 78.)

[59] B. Ghoshal and A. Tucker. Estimating uncertainty and interpretability in deep learning for coronavirus (covid-19) detection. *arXiv preprint arXiv:2003.10769*, 2020. (Cited on page 77.)

[60] A. Gillioz, J. Casas, E. Mugellini, and O. Abou Khaled. Overview of the transformer-based models for nlp tasks. In *FedCSIS*, pages 179–183. IEEE, 2020. (Cited on page 77.)

[61] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen. Slot-gated modeling for joint slot filling and intent prediction. In *NAACL-HLT*, pages 753–757, 2018. (Cited on page 86.)

[62] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. (Cited on page 83.)

[63] M. Gritta and I. Iacobacci. Xeroalign: Zero-shot cross-lingual transformer alignment. In *ACL Findings*, pages 371–381, 2021. (Cited on pages 54, 56, 66, and 76.)

[64] E. J. Gumbel. Statistical theory of extreme values and some practical applications. a series of lectures. *Number 33. US Govt. Print. Office*, 1954. (Cited on page 83.)

[65] J. Guo, D. J. Shah, and R. Barzilay. Multi-source domain adaptation with mixture of experts. In *EMNLP*, pages 4694–4703, 2018. (Cited on page 15.)

[66] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang. Transformer in transformer. In *NeurIPS*, pages 15908–15919, 2021. (Cited on pages 77 and 93.)

[67] M. Haspelmath. How hopeless is genealogical linguistics, and how advanced is areal linguistics? *Studies in Language*, 28(1):209–223, 2004. (Cited on page 55.)

[68] J. He, X. Zhang, S. Lei, Z. Chen, F. Chen, A. Alhamadani, B. Xiao, and C. Lu. Towards more accurate uncertainty estimation in text classification. In *EMNLP*, pages 8362–8372, 2020. (Cited on page 80.)

[69] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop*, 1990. (Cited on page 86.)

[70] M. Henderson. Machine learning for dialog state tracking: A review. *Proceedings of The First International Workshop on Machine Learning in Spoken Language Processing*, 2015. (Cited on page 1.)

[71] M. Henderson, I. Vulić, D. Gerz, I. Casanueva, P. Budzianowski, S. Coope, G. Spithourakis, T.-H. Wen, N. Mrkšić, and P.-H. Su. Training neural response selection for task-oriented dialogue systems. In *ACL*, pages 5392–5404, 2019. (Cited on page 29.)

[72] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017. (Cited on pages 81 and 86.)

[73] J. Herzig, M. Shmueli-Scheuer, T. Sandbank, and D. Konopnicki. Neural response generation for customer service based on personality traits. In *INLG*, pages 252–256, 2017. (Cited on page 32.)

[74] C.-J. Hoel, K. Wolff, and L. Laine. Tactical decision-making in autonomous driving by reinforcement learning with uncertainty estimation. In *IEEE Intelligent Vehicles Symposium*, pages 1563–1569. IEEE, 2020. (Cited on page 77.)

[75] 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*, 33:20179–20191, 2020.

(Cited on page 58.)

[76] M. B. Hoy. Alexa, siri, cortana, and more: an introduction to voice assistants. *Medical Reference Services Quarterly*, 37(1):81–88, 2018. (Cited on page 1.)

[77] D. A. Hudson and L. Zitnick. Generative adversarial transformers. In *ICML*, pages 4487–4499. PMLR, 2021. (Cited on page 93.)

[78] C.-C. Hung, A. Lauscher, I. Vulić, S. P. Ponzetto, and G. Glavaš. Multi2woz: A robust multilingual dataset and conversational pretraining for task-oriented dialog. In *NAACL-HLT*, number 3687–3703, 2022. (Cited on pages 50, 54, 55, and 56.)

[79] S. Hussain, O. Ameri Sianaki, and N. Ababneh. A survey on conversational agents/chatbots classification and design techniques. In *AINA Workshop*, pages 946–956. Springer, 2019. (Cited on page 1.)

[80] IGLU. Interactive grounded language understanding in a collaborative environment. https://www.iglu-contest.net/, 2022. (Cited on page 1.)

[81] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. (Cited on pages 79, 82, and 83.)

[82] D. Jannach, A. Manzoor, W. Cai, and L. Chen. A survey on conversational recommender systems. *ACM Computing Surveys*, 54(5):1–36, 2021. (Cited on page 1.)

[83] P. Jayarao and A. Srivastava. Intent detection for code-mix utterances in task oriented dialogue systems. In *ICEECCOT*, pages 583–587. IEEE, 2018. (Cited on page 56.)

[84] M. Jianjun, P. Jiahuan, and H. E. Degen. Identification of English functional noun phrases using CRFs combining the semantic information. *Journal of Chinese Information Processing*, 30(6):59–66, 2016. (Cited on page 55.)

[85] C. K. Joshi, F. Mi, and B. Faltings. Personalization in goal-oriented dialog. In *NeurIPS Workshop on Conversational AI*, 2017. (Cited on pages 5, 29, 30, 32, 33, 37, 39, 40, and 41.)

[86] A. Joulin, P. Bojanowski, T. Mikolov, H. Jégou, and É. Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *EMNLP*, pages 2979–2984, 2018. (Cited on page 76.)

[87] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi. Neural network-based uncertainty quantification: A survey of methodologies and applications. *IEEE Access*, 6:36218–36234, 2018. (Cited on pages 77 and 78.)

[88] P. Kaliamoorthi, A. Siddhant, E. Li, and M. Johnson. Distilling large language models into tiny and effective students using pqrnn. *arXiv preprint arXiv:2101.08890*, 2021. (Cited on page 54.)

[89] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, volume 30, pages 5574–5584, 2017. (Cited on page 81.)

[90] S. Kim, L. F. D'Haro, R. E. Banchs, J. D. Williams, M. Henderson, and K. Yoshino. The fifth dialog state tracking challenge. In *SLT Workshop*, pages 511–517. IEEE, 2016. (Cited on page 50.)

[91] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. (Cited on pages 21, 40, and 88.)

[92] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014. (Cited on page 78.)

[93] D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *NeurIPS*, volume 28, pages 2575–2583, 2015. (Cited on page 82.)

[94] N. Kitaev, Ł. Kaiser, and A. Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020. (Cited on page 93.)

[95] D. Kondratyuk. Cross-lingual lemmatization and morphology tagging with two-stage multilingual bert fine-tuning. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 12–18, 2019. (Cited on page 55.)

[96] J. Krishnan, A. Anastasopoulos, H. Purohit, and H. Rangwala. Multilingual code-switching for zero-shot cross-lingual intent prediction and slot filling. In *MRL Workshop*, pages 211–223, 2021. (Cited on page 54.)

[97] A. Kumar, P. Ku, A. Goyal, A. Metallinou, and D. Hakkani-Tur. Ma-dst: Multi-attention-based scalable dialog state tracking. In *AAAI*, volume 34, pages 8107–8114, 2020. (Cited on page 57.)

[98] T. M. Lai, Q. H. Tran, T. Bui, and D. Kihara. A simple but effective bert model for dialog state tracking on resource-limited systems. In *ICASSP*, pages 8034–8038. IEEE, 2020. (Cited on page 64.)

[99] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017. (Cited on pages 79, 80, and 87.)

[100] P. Le, M. Dymetman, and J.-M. Renders. LSTM-based mixture-of-experts for knowledge-aware dialogues. In *RepL4NLP Workshop*, pages 94–99, 2016. (Cited on page 15.)

[101] H. Lee, J. Lee, and T.-Y. Kim. Sumbt: Slot-utterance matching for universal and scalable belief tracking. In *ACL*, pages 5478–5483, 2019. (Cited on page 64.)

[102] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, pages 3744–3753. PMLR, 2019. (Cited on pages 79 and 93.)

[103] W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, and D. Yin. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *ACL*, pages 1437–1447, 2018. (Cited on page 14.)

[104] A. W. Li, V. Jiang, S. Y. Feng, J. Sprague, W. Zhou, and J. Hoey. Aloha: Artificial learning of human attributes for dialogue agents. In *AAAI*, pages 8155–8163, 2020. (Cited on pages 30 and 32.)

[105] H. Li, A. Arora, S. Chen, A. Gupta, S. Gupta, and Y. Mehdad. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. In *EACL*, pages 2950–2962, 2021. (Cited on pages 50, 54, and 56.)

[106] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan. A persona-based neural conversation model. In *ACL*, pages 994–1003, 2016. (Cited on pages 14, 29, and 31.)

[107] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky. Adversarial learning for neural dialogue generation. In *EMNLP*, pages 2157–2169, 2017. (Cited on page 14.)

[108] X. Li, G. Tur, D. Hakkani-Tür, and Q. Li. Personal knowledge graph population from user utterances in conversational understanding. In *SLT Workshop*, pages 224–229. IEEE, 2014. (Cited on pages 30 and 32.)

[109] T. Limisiewicz and D. Mareček. Syntax representation in word embeddings and neural networks – a survey. In *Proceedings of the 20th Conference ITAT 2020: Automata, Formal and Natural Languages Workshop*, 2020. (Cited on page 55.)

[110] G. Lin, D. W. Engel, and P. W. Eslinger. Survey and evaluate uncertainty quantification methodologies. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2012. (Cited on page 77.)

[111] Z. Lin, D. Cai, Y. Wang, X. Liu, H.-T. Zheng, and S. Shi. Grayscale data construction and multi-level ranking objective for dialogue response selection. *arXiv preprint arXiv:2004.02421*, 2020. (Cited on page 29.)

[112] Z. Lin, Z. Liu, G. I. Winata, S. Cahyawijaya, A. Madotto, Y. Bang, E. Ishii, and P. Fung. Xpersona: Evaluating multilingual personalized chatbot. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 102–112, 2021. (Cited on page 50.)

[113] Z. Lin, A. Madotto, G. I. Winata, P. Xu, F. Jiang, Y. Hu, C. Shi, and P. Fung. Bitod: A bilingual multi-domain dataset for task-oriented dialogue modeling. In *NeurIPS*, 2021. (Cited on pages 50 and 51.)

[114] B. Liu and I. R. Lane. End-to-end learning of task-oriented dialogs. In *NAACL-HLT Student Research Workshop*, pages 67–73, 2018. (Cited on page 1.)

[115] B. Liu, G. Tür, D. Hakkani-Tur, P. Shah, and L. Heck. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *NAACL-HLT*, pages 2060–2069, 2018. (Cited on page 2.)

[116] F. Liu and J. Perez. Gated end-to-end memory networks. In *EACL*, pages 1–10, 2017. (Cited on page 29.)

[117] Q. Liu, Y. Chen, B. Chen, J.-G. Lou, Z. Chen, B. Zhou, and D. Zhang. You impress me: Dialogue generation via mutual persona perception. In *ACL*, pages 1417–1427, 2020. (Cited on page 32.)

[118] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020. (Cited on page 51.)

[119] Z. Liu, J. Shin, Y. Xu, G. I. Winata, P. Xu, A. Madotto, and P. Fung. Zero-shot cross-lingual dialogue systems with transferable latent variables. In *EMNLP-IJCNLP*, pages 1297–1303, 2019. (Cited on pages 56 and 76.)

[120] Z. Liu, G. I. Winata, Z. Lin, P. Xu, and P. Fung. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. In *AAAI*, volume 34, pages 8433–8440, 2020. (Cited on pages 56, 75, and 76.)

[121] Z. Liu, G. I. Winata, S. Cahyawijaya, A. Madotto, Z. Lin, and P. Fung. On the importance of word order information in cross-lingual sequence labeling. In *AAAI*, volume 35, pages 13461–13469, 2021. (Cited on pages 56, 66, and 76.)

[122] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. (Cited on page 63.)

[123] S. Louvan and B. Magnini. Simple is better! lightweight data augmentation for low resource slot filling and intent classification. In *PACLIC*, pages 167–177, 2020. (Cited on page 54.)

[124] Y. Lu, M. Srivastava, J. Kramer, H. Elfardy, A. Kahn, S. Wang, and V. Bhardwaj. Goal-oriented end-to-end conversational models with profile features in a real-world setting. In *NAACL-HLT*, pages 48–55, 2019. (Cited on page 29.)

[125] Y. Luan, C. Brockett, B. Dolan, J. Gao, and M. Galley. Multi-task learning for speaker-role adaptation in neural conversation models. In *IJCNLP*, pages 605–614, 2017. (Cited on page 32.)

[126] L. Luo, W. Huang, Q. Zeng, Z. Nie, and X. Sun. Learning personalized end-to-end goal-oriented dialog. In *AAAI*, pages 6794–6801, 2019. (Cited on pages 5, 29, 30, 31, 32, 33, 35, 37, 39, 40, and 41.)

[127] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, pages 1412–1421, 2015. (Cited on page 17.)

[128] J. Ma, J. Pei, D. Huang, and D. Song. Syntactic parsing of clause constituents for statistical machine translation. *International Journal of Computational Science and Engineering*, 17(1):126–132, 2018. (Cited on page 55.)

[129] L. Ma, M. Li, W.-N. Zhang, J. Li, and T. Liu. Unstructured text enhanced open-domain dialogue system: A systematic survey. *ACM Transactions on Information Systems*, 40(1):1–44, 2021. (Cited on page 1.)

[130] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *NAACL-HLT*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. (Cited on page 86.)

[131] B. MacWhinney, J. F. Kroll, et al. A unified model of language acquisition. *Handbook of bilingualism: Psycholinguistic approaches*, 4967:50–70, 2005. (Cited on page 55.)

[132] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017. (Cited on pages 79, 82, and 83.)

[133] A. Madotto, C.-S. Wu, and P. Fung. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *ACL*, pages 1468–1478, 2018. (Cited on page 29.)

[134] A. Madotto, Z. Lin, C.-S. Wu, and P. Fung. Personalizing dialogue agents via meta-learning. In *ACL*, pages 5454–5459, 2019. (Cited on page 32.)

[135] A. Martin, C. Ollion, F. Strub, S. L. Corff, and O. Pietquin. The monte carlo transformer: a stochastic self-attention model for sequence prediction. *arXiv preprint arXiv:2007.08620*, 2020. (Cited on page 80.)

[136] S. Masoudnia and R. Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014. (Cited on pages 3, 12, and 14.)

[137] B. W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975. (Cited on page 87.)

[138] P.-E. Mazare, S. Humeau, M. Raison, and A. Bordes. Training millions of personalized dialogue agents. In *EMNLP*, pages 2775–2779, 2018. (Cited on pages 31 and 32.)

[139] B. McCann, J. Bradbury, C. Xiong, and R. Socher. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30, 2017. (Cited on pages 66 and 76.)

[140] L. McInnes, J. Healy, N. Saul, and L. Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. (Cited on pages 51 and 67.)

[141] D. P. Medeiros. Ultra: Universal grammar as a universal parser. *Frontiers in Psychology*, 9:155, 2018. (Cited on page 55.)

[142] S. Mehri, T. Srinivasan, and M. Eskenazi. Structured fusion networks for dialog. In *SIGDIAL*, pages 165–177, 2019. (Cited on pages 20, 21, 22, and 23.)

[143] F. Mi, M. Huang, J. Zhang, and B. Faltings. Meta-learning for low-resource natural language generation in task-oriented dialogue systems. In *IJCAI*, pages 3151–3157, 2019. (Cited on page 13.)

[144] A. Miller, W. Feng, D. Batra, A. Bordes, A. Fisch, J. Lu, D. Parikh, and J. Weston. Parlai: A dialog research software platform. In *EMNLP System Demonstrations*, pages 79–84, 2017. (Cited on page 1.)

[145] K. Mo, Y. Zhang, S. Li, J. Li, and Q. Yang. Personalizing a dialogue system with transfer reinforcement learning. In *AAAI*, pages 5317–5324, 2018. (Cited on pages 29 and 32.)

[146] N. Mrkšić and I. Vulić. Fully statistical neural belief tracking. In *ACL*, pages 108–113, 2018. (Cited on pages 5, 50, and 64.)

[147] N. Mrkšić, D. O. Séaghdha, T.-H. Wen, B. Thomson, and S. Young. Neural belief tracker: Data-driven dialogue state tracking. In *ACL*, pages 1777–1788, 2017. (Cited on pages 1 and 64.)

[148] N. Mrkšić, I. Vulić, D. Ó. Séaghdha, I. Leviant, R. Reichart, M. Gašić, A. Korhonen, and S. Young. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324, 2017. (Cited on pages 5, 50, 51, 53, 54, 56, 61, 62, 64, and 96.)

[149] A. Müller, S. Wichmann, V. Velupillai, C. H. Brown, P. Brown, S. Sauppe, E. W. Holman, D. Bakker,

J.-M. List, D. Egorov, et al. ASJP world language tree of lexical similarity: Version 3 (July 2010). `https://asjp.clld.org/static/WorldLanguageTree-003.pdf`, 2010. (Cited on page 55.)

[150] T. Naseem, B. Snyder, J. Eisenstein, and R. Barzilay. Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of Artificial Intelligence Research*, 36:341–385, 2009. (Cited on page 55.)

[151] J. Ni, T. Young, V. Pandelea, F. Xue, V. Adiga, and E. Cambria. Recent advances in deep learning based dialogue systems: A systematic survey. *Artificial Intelligence Review*, (1–101), 2022. (Cited on pages 1, 49, and 55.)

[152] J. Nivre. Towards a universal grammar for natural language processing. In *CICLing*, pages 3–16. Springer, 2015. (Cited on page 55.)

[153] E. Nouri and E. Hosseini-Asl. Toward scalable neural dialogue state tracking model. In *NeurIPS*, 2018. (Cited on page 64.)

[154] J. Nouri and R. Yangarber. From alignment of etymological data to phylogenetic inference via population genetics. In *CogACLL Workshop*, pages 27–37, 2016. (Cited on page 55.)

[155] M. Nuruzzaman and O. K. Hussain. A survey on chatbot implementation in customer service industry through deep neural networks. In *ICEBE*, pages 54–61. IEEE, 2018. (Cited on page 1.)

[156] N. Oco, L. R. Syliongka, R. E. Roxas, and J. Ilao. Dice's coefficient on trigram profiles as metric for language similarity. In *O-COCOSDA/CASLRE*, pages 1–4. IEEE, 2013. (Cited on page 55.)

[157] G. Oh and Y. S. Hong. Managing market risk caused by customer preference uncertainty in product family design with launch flexibility: Product option strategy. *Computers & Industrial Engineering*, 151:106975, 2021. (Cited on page 77.)

[158] K.-J. Oh, J. Song, H. An, and H.-J. Choi. A framework of callbot for dialogue process in seperated network. In *BigComp*, pages 362–364. IEEE, 2021. (Cited on page 1.)

[159] H. O'Horan, Y. Berzak, I. Vulić, R. Reichart, and A. Korhonen. Survey on the use of typological information in natural language processing. In *COLING Technical Papers*, pages 1297–1308, 2016. (Cited on page 55.)

[160] S. Panda, C. Tirkaz, T. Falke, and P. Lehnen. Multilingual paraphrase generation for bootstrapping new features in task-oriented dialog systems. In *Workshop on NLP for Conversational AI*, pages 30–39, 2021. (Cited on page 55.)

[161] H. H. Park, K. J. Zhang, C. Haley, K. Steimel, H. Liu, and L. Schwartz. Morphology matters: A multilingual language modeling analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276, 2021. (Cited on page 55.)

[162] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *ICML*, pages 1310–1318, 2013. (Cited on pages 21 and 40.)

[163] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*, pages 8024–8035, 2019. (Cited on page 88.)

[164] N. D. Pattengale, E. J. Gottlieb, and B. M. Moret. Efficiently computing the robinson-foulds metric. *Journal of computational biology*, 14(6):724–735, 2007. (Cited on page 61.)

[165] M. Paul, A. Finch, and E. Sumita. How to choose the best pivot language for automatic translation of low-resource languages. *ACM Transactions on Asian Language Information Processing (TALIP)*, 12 (4):1–17, 2013. (Cited on page 50.)

[166] J. Pei, P. Ren, and M. de Rijke. A modular task-oriented dialogue system using a neural mixture-of-experts. In *SIGIR Workshop on Conversational Interaction Systems*, 2019. (Cited on pages 3, 11, 15, 29, and 51.)

[167] J. Pei, A. Stienstra, J. Kiseleva, and M. de Rijke. SEntNet: Source-aware recurrent entity network for dialogue response selection. In *IJCAI Workshop SCAI*, 2019. (Cited on pages 3, 12, 29, and 33.)

[168] J. Pei, P. Ren, C. Monz, and M. de Rijke. Retrospective and prospective mixture-of-generators for task-oriented dialogue response generation. In *ECAI*, pages 2148–2155, 2020. (Cited on pages 3, 11, 29, and 51.)

[169] J. Pei, P. Ren, and M. de Rijke. A cooperative memory network for personalized task-oriented dialogue systems with incomplete user profiles. In *The Web Conference*, pages 1552–1561, 2021. (Cited on pages 3, 29, and 54.)

[170] J. Pei, C. Wang, and G. Szarvas. Transformer uncertainty estimation with hierarchical stochastic attention. In *AAAI*, pages 11147–11155, 2022. (Cited on pages 3 and 77.)

[171] J. Pei, G. Yan, P. Ren, and M. de Rijke. Mixture-of-languages routing for multilingual task-oriented dialogue systems. *Under review*, 2022. (Cited on pages 3 and 49.)

[172] A. Potapczynski, G. Loaiza-Ganem, and J. P. Cunningham. Invertible gaussian reparameterization: revisiting the gumbel-softmax. In *NeurIPS*, pages 12311–12321, 2020. (Cited on page 79.)

[173] Q. Qian, M. Huang, H. Zhao, J. Xu, and X. Zhu. Assigning personality/profile to a chatting machine for coherent conversation generation. In *IJCAI*, pages 4279–4285, 2018. (Cited on page 32.)

[174] L. Qin, M. Ni, Y. Zhang, and W. Che. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual nlp. In *IJCAI*, pages 3853–3860, 2021. (Cited on pages 54, 56, 75, and 76.)

[175] J. Quan, S. Zhang, Q. Cao, Z. Li, and D. Xiong. Risawoz: A large-scale multi-domain wizard-of-oz dataset with rich semantic annotations for task-oriented dialogue modeling. In *EMNLP*, pages 930–940, 2020. (Cited on page 1.)

[176] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. (Cited on page 54.)

[177] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020. (Cited on page 59.)

[178] J. Rajendran, J. Ganhotra, S. Singh, and L. Polymenakos. Learning end-to-end goal-oriented dialog with multiple answers. In *EMNLP*, pages 3834–3843, 2018. (Cited on page 33.)

[179] A. Rastogi, D. Hakkani-Tür, and L. Heck. Scalable multi-domain dialogue state tracking. In *ASRU Workshop*, pages 561–568. IEEE, 2017. (Cited on page 1.)

[180] A. Rastogi, R. Gupta, and D. Hakkani-Tur. Multi-task learning for joint language understanding and dialogue state tracking. In *SIGDIAL*, pages 376–384, 2018. (Cited on page 13.)

[181] E. Razumovskaia, G. Glavaš, O. Majewska, A. Korhonen, and I. Vulic. Crossing the conversational chasm: A primer on multilingual task-oriented dialogue systems. *CoRR*, 2021. (Cited on pages 5, 49, 51, 56, and 60.)

[182] E. Razumovskaia, G. Glavaš, O. Majewska, E. Ponti, and I. Vulić. Natural language processing for multilingual task-oriented dialogue. In *ACL Tutorial Abstracts*, pages 44–50, 2022. (Cited on page 49.)

[183] E. Razumovskaia, I. Vulić, and A. Korhonen. Data augmentation and learned layer aggregation for improved multilingual language understanding in dialogue. In *ACL Findings*, pages 2017–2033, 2022. (Cited on pages 54 and 66.)

[184] L. Ren, K. Xie, L. Chen, and K. Yu. Towards universal dialogue state tracking. In *EMNLP*, pages 2780–2786, 2018. (Cited on page 64.)

[185] P. Ren, Z. Chen, Z. Ren, E. Kanoulas, C. Monz, and M. De Rijke. Conversations with search engines: Serp-based conversational response generation. *ACM Transactions on Information Systems*, 39(4): 1–29, 2021. (Cited on page 77.)

[186] S. Riedmaier, B. Danquah, B. Schick, and F. Diermeyer. Unified framework and survey for model verification, validation and uncertainty quantification. *Archives of Computational Methods in Engineering*, 28(4):2655–2688, 2021. (Cited on page 77.)

[187] S. Ruder, I. Vulić, and A. Søgaard. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631, 2019. (Cited on page 55.)

[188] SCAI. Search-oriented conversational ai. `https://scai.info`, 2022. (Cited on page 1.)

[189] S. Scalise, E. Magni, and A. Bisetto. *Universals of language today*. Springer, 2009. (Cited on page 55.)

[190] S. Schuster, S. Gupta, R. Shah, and M. Lewis. Cross-lingual transfer learning for multilingual task oriented dialog. In *NAACL-HLT*, pages 3795–3805, 2019. (Cited on pages 50, 53, 56, 61, 63, 66, 76, and 96.)

[191] P. Schwab, D. Miladinovic, and W. Karlen. Granger-causal attentive mixtures of experts: Learning important features with neural networks. In *AAAI*, pages 4846–4853, 2019. (Cited on page 17.)

[192] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784, 2016. (Cited on pages 1 and 14.)

[193] I. V. Serban, R. Lowe, P. Henderson, L. Charlin, and J. Pineau. A survey of available corpora for building data-driven dialogue systems. *Dialogue & Discourse*, 9(1):1–49, 2018. (Cited on pages 50 and 55.)

[194] M. Serva and F. Petroni. Indo-european languages tree by levenshtein distance. *EPL (Europhysics Letters)*, 81(6):68005, 2008. (Cited on page 55.)

[195] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017. (Cited on page 17.)

[196] A. Shelmanov, E. Tsymbalov, D. Puzyrev, K. Fedyanin, A. Panchenko, and M. Panov. How certain is

your transformer? In *EACL*, pages 1833–1840, 2021. (Cited on page 80.)

[197] M. Shum, S. Zheng, W. Kryściński, C. Xiong, and R. Socher. Sketch-fill-ar: A persona-grounded chit-chat generation framework. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 118–131, 2020. (Cited on page 32.)

[198] A. Siddhant, M. Johnson, H. Tsai, N. Ari, J. Riesa, A. Bapna, O. Firat, and K. Raman. Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. In *AAAI*, volume 34, pages 8854–8861, 2020. (Cited on page 56.)

[199] A. Søgaard, I. Vulić, S. Ruder, and M. Faruqui. Cross-lingual word embeddings. *Synthesis Lectures on Human Language Technologies*, 12(2):1–132, 2019. (Cited on page 55.)

[200] H. Song, W.-N. Zhang, Y. Cui, D. Wang, and T. Liu. Exploiting persona information for diverse generation of conversational responses. In *IJCAI*, pages 5190–5196, 2019. (Cited on page 32.)

[201] H. Song, Y. Wang, W.-N. Zhang, X. Liu, and T. Liu. Generate, delete and rewrite: A three-stage framework for improving persona consistency of dialogue generation. In *ACL*, pages 5821–5831, 2020. (Cited on page 32.)

[202] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan. A neural network approach to context-sensitive generation of conversational responses. In *NAACL-HLT*, pages 196–205, 2015. (Cited on page 14.)

[203] G. P. Spithourakis, I. Vulić, M. Lis, I. Casanueva, and P. Budzianowski. Evi: Multilingual spoken dialogue tasks and dataset for knowledge-based enrolment, verification, and identification. *arXiv preprint arXiv:2204.13496*, 2022. (Cited on page 50.)

[204] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. (Cited on pages 79 and 87.)

[205] H. Sugiyama, T. Meguro, R. Higashinaka, and Y. Minami. Open-domain utterance generation for conversational dialogue systems using web-scale dependency structures. In *SIGDIAL*, pages 334–338, 2013. (Cited on page 1.)

[206] S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus, et al. End-to-end memory networks. In *NeurIPS*, pages 2440–2448, 2015. (Cited on pages 32 and 37.)

[207] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NeurIPS*, pages 3104–3112, 2014. (Cited on pages 31 and 32.)

[208] C. Tang and V. J. van Heuven. Mutual intelligibility and similarity of chinese dialects: Predicting judgments from objective measures. *Linguistics in the Netherlands*, 24(1):223–234, 2007. (Cited on page 55.)

[209] I. V. Tetko, P. Karpov, R. Van Deursen, and G. Godin. State-of-the-art augmented nlp transformer models for direct and single-step retrosynthesis. *Nature Communications*, 11(1):1–11, 2020. (Cited on page 77.)

[210] S. A. Thompson, R. E. Longacre, S. J. J. Hwang, and T. Shopen. *Language typology and syntactic description*. Cambridge University Press, 2007. (Cited on page 54.)

[211] A. Tigunova. Extracting personal information from conversations. In *The Web Conference: Companion*, pages 284–288, 2020. (Cited on pages 30 and 32.)

[212] A. Tigunova, A. Yates, P. Mirza, and G. Weikum. Listening between the lines: learning personal attributes from conversations. In *The Web Conference*, pages 1818–1828. ACM, 2019. (Cited on pages 30 and 32.)

[213] S. Upadhyay, M. Faruqui, G. Tür, H.-T. Dilek, and L. Heck. (almost) zero-shot cross-lingual spoken language understanding. In *ICASSP*, pages 6034–6038. IEEE, 2018. (Cited on pages 50, 56, and 76.)

[214] P. N. Van, T. C. Hoang, D. N. Manh, Q. N. Minh, and L. T. Quoc. Viwoz: A multi-domain task-oriented dialogue systems dataset for low-resource language. *arXiv preprint arXiv:2203.07742*, 2022. (Cited on pages 50, 54, and 56.)

[215] P. G. J. van Sterkenburg. *Unity and diversity of languages*. John Benjamins Publishing, 2008. (Cited on pages 54 and 55.)

[216] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. (Cited on pages 6, 77, 81, and 87.)

[217] O. Vinyals and Q. Le. A neural conversational model. In *ICML Deep Learning Workshop*, 2015. (Cited on page 14.)

[218] I. Vulić, N. Mrkšić, R. Reichart, D. Ó. Séaghdha, S. Young, and A. Korhonen. Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules. In *ACL*, pages 56–68, 2017. (Cited on page 64.)

[219] A. Vyas, A. Katharopoulos, and F. Fleuret. Fast transformers with clustered attention. In *NeurIPS*,

2020. (Cited on pages 79, 84, and 85.)

[220] B. Wang, L. Shang, C. Lioma, X. Jiang, H. Yang, Q. Liu, and J. G. Simonsen. On position embeddings in bert. In *ICLR*, 2020. (Cited on page 77.)

[221] B. Wang, Q. Xie, J. Pei, P. Tiwari, Z. Li, and J. Fu. Pre-trained language models in biomedical domain: a systematic survey. *Association for Computing Machinery*, 2021. (Cited on page 1.)

[222] C. Wang and M. Niepert. State-regularized recurrent neural networks. In *ICML*, pages 6596–6606. PMLR, 2019. (Cited on pages 79 and 84.)

[223] C. Wang, C. Lawrence, and M. Niepert. Uncertainty estimation and calibration with finite-state probabilistic rnns. In *ICLR*, 2021. (Cited on page 79.)

[224] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *SIGKDD*, pages 1235–1244. ACM, 2015. (Cited on page 30.)

[225] J. Wang and Y. Dong. Measurement of text similarity: a survey. *Information*, 11(9):421, 2020. (Cited on page 55.)

[226] Q. Wang and H. Van Hoof. Doubly stochastic variational inference for neural processes with hierarchical latent variables. In *ICML*, pages 10018–10028. PMLR, 2020. (Cited on page 81.)

[227] A. Warstadt, A. Singh, and S. R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. (Cited on page 86.)

[228] H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han. A survey of joint intent detection and slot-filling models in natural language understanding. *ACM Computing Surveys*, 2021. (Cited on pages 1 and 2.)

[229] T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gasic, L. M. R. Barahona, P.-H. Su, S. Ultes, and S. Young. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*, pages 438–449, 2017. (Cited on pages 1, 11, 14, 33, and 61.)

[230] L. J. Whaley. *Introduction to typology: The unity and diversity of language*. SAGE publications, 1996. (Cited on page 55.)

[231] J. D. Williams, A. Raux, and M. Henderson. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33, 2016. (Cited on page 1.)

[232] J. D. Williams, K. Asadi, and G. Zweig. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *ACL*, pages 665–677, 2017. (Cited on pages 1, 11, and 29.)

[233] T. Wolf, V. Sanh, J. Chaumond, and C. Delangue. Transfertransfo: A transfer learning approach for neural network based conversational agents. In *NeurIPS 2018 CAI Workshop*, 2018. (Cited on page 32.)

[234] C.-S. Wu, A. Madotto, G. I. Winata, and P. Fung. End-to-end dynamic query memory network for entity-value independent task-oriented dialog. In *ICASSP*, pages 6154–6158, 2018. (Cited on page 29.)

[235] 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. In *ACL*, pages 808–819, 2019. (Cited on page 57.)

[236] C.-S. Wu, A. Madotto, Z. Lin, P. Xu, and P. Fung. Getting to know you: User attribute extraction from dialogues. In *LREC*, pages 581–589, 2020. (Cited on page 30.)

[237] S. Wu and M. Dredze. Are all languages created equal in multilingual bert? In *RepL4NLP Workshop*, pages 120–130, 2020. (Cited on page 5.)

[238] L. Xiang, J. Zhu, Y. Zhao, Y. Zhou, and C. Zong. Robust cross-lingual task-oriented dialogue. *Transactions on Asian and Low-Resource Language Information Processing*, 20(6):1–24, 2021. (Cited on page 56.)

[239] M. Xu, P. Li, H. Yang, P. Ren, Z. Ren, Z. Chen, and J. Ma. A neural topical expansion framework for unstructured persona-oriented dialogue generation. In *ECAI*, pages –, 2020. (Cited on page 32.)

[240] P. Xu and Q. Hu. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *ACL*, pages 1448–1457, 2018. (Cited on page 57.)

[241] W. Xu, B. Haider, and S. Mansour. End-to-end slot alignment and recognition for cross-lingual nlu. In *EMNLP*, pages 5052–5063, 2020. (Cited on pages 55 and 56.)

[242] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. In *NAACL-HLT*, 2021. (Cited on pages 6, 50, 51, 58, and 96.)

[243] G. Yan, J. Pei, P. Ren, Z. Ren, X. Xin, H. Liang, M. de Rijke, and Z. Chen. ReMeDi: Resources for multi-domain, multi-service, medical dialogues. In *SIGIR*, pages 3013–3024, 2022. (Cited on pages 1, 5, 54, and 58.)

[244] Z. Yan, N. Duan, P. Chen, M. Zhou, J. Zhou, and Z. Li. Building task-oriented dialogue systems for online shopping. In *AAAI*, pages 4618–4626, 2017. (Cited on pages 1 and 14.)

[245] M. Yang, Z. Zhao, W. Zhao, X. Chen, J. Zhu, L. Zhou, and Z. Cao. Personalized response generation

via domain adaptation. In *SIGIR*, pages 1021–1024, 2017. (Cited on page 32.)

[246] X. Yang, Y.-N. Chen, D. Hakkani-Tür, P. Crook, X. Li, J. Gao, and L. Deng. End-to-end joint learning of natural language understanding and dialogue manager. In *ICASSP*, pages 5690–5694. IEEE, 2017. (Cited on page 2.)

[247] S. Yi, R. Goel, C. Khatri, T. Chung, B. Hedayatnia, A. Venkatesh, R. Gabriel, and D. Hakkani-Tur. Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 65–75, 2019. (Cited on page 13.)

[248] Y. Yin, L. Shang, X. Jiang, X. Chen, and Q. Liu. Dialog state tracking with reinforced data augmentation. In *AAAI*, volume 34, pages 9474–9481, 2020. (Cited on pages 54 and 64.)

[249] S. Young, M. Gašić, B. Thomson, and J. D. Williams. POMDP-based statistical spoken dialog systems: A review. *IEEE*, 101(5):1160–1179, 2013. (Cited on pages 1, 11, 14, and 29.)

[250] W. Zeng, A. Abuduweili, L. Li, and P. Yang. Automatic generation of personalized comment based on user profile. In *ACL Student Research Workshop*, pages 229–235, 2019. (Cited on page 29.)

[251] B. Zhang, X. Xu, X. Li, Y. Ye, X. Chen, and Z. Wang. A memory network based end-to-end personalized task-oriented dialogue generation. *Knowledge-Based Systems*, page 106398, 2020. (Cited on pages 5, 29, 30, 32, 33, 40, and 41.)

[252] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *ACL*, pages 2204–2213, 2018. (Cited on pages 29, 31, and 32.)

[253] W.-N. Zhang, Q. Zhu, Y. Wang, Y. Zhao, and T. Liu. Neural personalized response generation as domain adaptation. *World Wide Web*, 22(4):1427–1446, 2019. (Cited on page 32.)

[254] Z. Zhang, M. Huang, Z. Zhao, F. Ji, H. Chen, and X. Zhu. Memory-augmented dialogue management for task-oriented dialogue systems. *ACM Transactions on Information Systems*, 37(3):34, 2019. (Cited on pages 1 and 13.)

[255] Z. Zhang, R. Takanobu, Q. Zhu, M. Huang, and X. Zhu. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027, 2020. (Cited on page 55.)

[256] T. Zhao, K. Xie, and M. Eskenazi. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *NAACL*, pages 1208–1218, 2019. (Cited on pages 11, 20, and 22.)

[257] Z. Zhao, S. Zhu, and K. Yu. Data augmentation with atomic templates for spoken language understanding. In *EMNLP-IJCNLP*, pages 3637–3643, 2019. (Cited on page 54.)

[258] Y. Zheng, G. Chen, M. Huang, S. Liu, and X. Zhu. Personalized dialogue generation with diversified traits. *arXiv preprint arXiv:1901.09672*, 2019. (Cited on page 32.)

[259] Y. Zheng, R. Zhang, M. Huang, and X. Mao. A pre-training based personalized dialogue generation model with persona-sparse data. In *AAAI*, pages 9693–9700, 2020. (Cited on page 32.)

[260] V. Zhong, C. Xiong, and R. Socher. Global-locally self-attentive encoder for dialogue state tracking. In *ACL*, pages 1458–1467, 2018. (Cited on pages 11, 13, and 64.)

[261] H. Zhou, I. Iacobacci, and P. Minervini. Xqa-dst: Multi-domain and multi-lingual dialogue state tracking. *arXiv preprint arXiv:2204.05895*, 2022. (Cited on pages 51, 54, 56, 64, and 75.)

[262] L. Zhou, J. Gao, D. Li, and H.-Y. Shum. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93, 2020. (Cited on page 32.)

[263] Q. Zhu, Z. Zhang, Y. Fang, X. Li, R. Takanobu, J. Li, B. Peng, J. Gao, X. Zhu, and M. Huang. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. In *ACL System Demonstrations*, pages 142–149, 2020. (Cited on page 1.)

[264] L. Zuo, K. Qian, B. Yang, and Z. Yu. Allwoz: Towards multilingual task-oriented dialog systems for all. *arXiv preprint arXiv:2112.08333*, 2021. (Cited on pages 54, 56, and 57.)

# Summary

Dialogue systems (a.k.a. conversational agents) aim to help people interact with machines through natural language. They are playing an increasingly important role in our daily life. Unlike chitchat, task-oriented dialogue systems focus on accurately assisting people to achieve specific goals (e.g., booking restaurants or tickets, scheduling meetings, providing medical services). Besides plain text, task-oriented dialogue systems form utterances with predefined goal-related semantic constraints (e.g., intents, slots, states, actions, frames). There are two categories of approaches: modularized pipeline agents and end-to-end single-module agents. A challenge of the former is error accumulation because multiple modules are sequentially dependent. And concerning the latter, it is impractical to use a single general agent to handle all complex cases. In this thesis, we introduce a new framework for TDSs, namely *collaborative task-oriented dialogue systems*. Within this framework, we have proposed a series of approaches where a group of collaborative specialized agents outperforms a single general agent.

The thesis focuses on four dimensions of collaborative dialogue agents: (i) Model collaboration: we have proposed a mixture-of-generators network, where the chair generator is able to collaborate a mixture of expert generators for generating high-quality responses. (ii) User collaboration: we have devised a cooperative memory network, where a user collaborates with a mixture of neighboring users to enrich incomplete profiles and enhance response selection. (iii) Language collaboration: we have introduced a mixture-of-languages routing model, where a language route can be influenced by multiple other language routes for global optimization in multilingual task-oriented dialogue systems. (iv) Uncertainty estimation: we have reformulated transformers as sequential collaborative agents and studied uncertainty estimation towards trustworthy collaborative task-oriented dialogue systems.

Our main findings concern three key factors: (i) Partition aspects of collaborative agents play a vital role. We have explored how to partition dialogue agents in terms of aspects such as information sources, models, users, and languages. These partition aspects have been proven effective in our empirical studies. However, different partition aspects have a dramatic influence on the final performance. For example, the generated responses vary greatly depending on domain or system action. (ii) Collaborative agents should be equipped with appropriate collaboration mechanisms. Retrospective and prospective mixture-of-generators fit specifically with response generation. Hierarchical stochastic attention is only designed for models based on hierarchical attention. Incremental collaborative filtering can help with models with cooperative interactions. Mixture-of-languages routing is effective for models with multiple routes. (iii) Topological structures of collaborative agents vary a lot. We have explored how to connect different types of dialogue agents, i.e., sequentially or with different chair-expert setups.

As to future work, we call for research in three directions: (i) Self-supervised partition and collaboration, which aims to automatically learn the partitions and collaboration policies. (ii) Uncertainty estimation in collaborative agents, which aims to enable deterministic collaborative agents to estimate uncertainty while preserving their predictive performance; and (iii) Collaboration effectiveness and efficiency, which aims to improve the efficiency of collaboration while preserving the effectiveness of collaborative agents.

# Samenvatting

Dialoogsystemen (ook wel “conversational agents” genoemd) zijn bedoeld om mensen te helpen communiceren met machines door middel van natuurlijke taal. Ze spelen een steeds belangrijkere rol in ons dagelijks leven. In tegenstelling tot chatbots, zijn taakgerichte dialoogsystemen gericht op het nauwkeurig helpen van mensen om specifieke doelen te bereiken (bijvoorbeeld het boeken van restaurants of tickets, het plannen van vergaderingen, het verlenen van medische diensten). Naast platte tekst produceren taakgeoriënteerde dialoogsystemen uitingen met behulp van vooraf gedefinieerde doelgerelateerde semantische beperkingen (bijv. intenties, slots, toestanden, acties, frames). Er zijn twee categorieën benaderingen: gemodulariseerde *pipeline*agenten en *end-to-end* agenten bestaande uit een enkele module. Een uitdaging van de eerste is de accumulatie van fouten omdat meerdere modules afhankelijk zijn van elkaar. En wat dat laatste betreft, is het niet praktisch om één enkele algemene agent in te zetten om alle complexe taken af te handelen. In dit proefschrift introduceren we een nieuw raamwerk voor TDS’en, namelijk collaboratieve taakgeoriënteerde dialoogsystemen. Binnen dit kader hebben we een reeks benaderingen voorgesteld waarbij een groep samenwerkende gespecialiseerde agenten beter presteert dan een enkele algemene agent.

Het proefschrift richt zich op vier dimensies van collaboratieve dialoogagenten: (i) Modelsamenwerking: we hebben een mengeling-van-generatorennetwerk voorgesteld, waarbij de voorzitter-generator in staat is om samen te werken met een mengeling van deskundige generatoren voor het genereren van hoogwaardige reacties. (ii) Gebruikerssamenwerking: we hebben een coöperatief geheugennetwerk bedacht, waarbij een gebruiker samenwerkt met een mix van naburige gebruikers om onvolledige profielen te verrijken en de responsselectie te verbeteren. (iii) Taalsamenwerking: we hebben een routemodel voor meerdere talen geïntroduceerd, waarbij een taalroute kan worden beïnvloed door meerdere andere taalroutes voor globale optimalisatie in meertalige taakgeoriënteerde dialoogsystemen. (iv) Onzekerheidsschatting: we hebben transformers geherformuleerd als sequentiële samenwerkingsagenten en onzekerheidsschatting bestudeerd die gericht zijn op betrouwbare collaboratieve taakgeoriënteerde dialoogsystemen.

Onze belangrijkste bevindingen hebben betrekking op drie sleutelfactoren: (i) Partitieaspecten van samenwerkende agenten spelen een cruciale rol. We hebben onderzocht hoe dialoogagenten kunnen worden ingedeeld in aspecten als informatiebronnen, modellen, gebruikers en talen. Deze aspecten zijn in onze empirische studies effectief gebleken. Verschillende partitieaspecten hebben echter een dramatische invloed op de uiteindelijke uitvoering. De gegenereerde reacties variëren bijvoorbeeld sterk, afhankelijk van domein- of systeemactie. (ii) Samenwerkende agenten moeten worden uitgerust met passende samenwerkingsmechanismen. Retrospectieve en prospectieve mix-van–generatoren passen specifiek bij het genereren van respons. Hiërarchische stochastische aandacht is alleen bedoeld voor modellen die gebaseerd zijn op hiërarchische aandacht. Incrementele *collaborative filtering* kan helpen met modellen met coperatieve interacties. Routing van verschillende talen is effectief voor modellen met meerdere routes. (iii) Topologische structuren van samenwerkende agenten variëren sterk. We hebben onderzocht hoe we verschillende soorten dialoogagenten kunnen verbinden, d.w.z. sequentieel of met verschillende voorzitter-expert-opstellingen.

Wat toekomstig werk betreft, pleiten we voor onderzoek in drie richtingen: (i) Zelf-gecontroleerde partitie en samenwerking, een richting die erop gericht is om automatisch de partities en het samenwerkingsbeleid te leren; (ii) Schatting van onzekerheid bij samenwerkende agenten, een richting die tot doel heeft om deterministische samenwerkende agenten in staat te stellen onzekerheid in te schatten terwijl hun voorspellende prestaties behouden blijven; en (iii) Effectiviteit en efficiëntie van samenwerking, een richting die tot doel heeft om de efficiëntie van samenwerking te verbeteren met behoud van de effectiviteit van samenwerkende agenten.