Data-Driven Policy Optimisation for Multi-Domain Task-Oriented Dialogue

Paweł Budzianowski

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Clare Hall College September 2020

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgements. This dissertation contains roughly 41,688 words including appendices, bibliography, footnotes and equations and has 37 figures and tables. Some of the material included in this thesis has been published in the Special Interest Group on Discourse and Dialogue (Budzianowski et al., 2017), the International Conference on Acoustics, Speech, and Signal Processing (Tegho et al., 2018), the Empirical Methods on Natural Language Processing (Budzianowski et al., 2018b), the 3rd Workshop on Neural Generation and Translation (Budzianowski and Vulić, 2019).

Paweł Budzianowski
September 2020

Acknowledgements

First, I would like to thank my supervisors, Anna Korhonen and Rich Turner. I really appreciate your help and guidance until the finish line. This PhD journey would not have been possible without Milica Gašić that gave me a chance to join a wonderful research group! Thanks to you I learnt how to plan and execute research, persevere with experiments and do not give up.

My dear HGB crew - it is hard to emphasize how much I learnt from you. Nikola taught me rich Serbian history and that you can do a lot with your phone, Eddy whose compassion and joyfulness made my first year PhD a bliss and a dream. Iñigo - your criticism made the whole journey worth pursuing. It is hard to describe how much I appreciate Shawn's generosity with ideas and guidance. Yen and Andy - I cannot believe I say this but working in the lab and living together was pretty spectacular!

The immense support from my college and my fellow Bartonians cannot be forgotten here. Thanks Cat, Chloe, Rashel, Emma and Alex. My forever gratitude goes to Abby for helping me survive (especially mentally) during this bumpy road. Your American positive spirit always cheered me up! To a second dear Californian, Pauline - I will never forget the American experience and all our road trips. Big thanks go to the CUVC club and my beloved captains Jirka and Max for a magical 2018 Varsity win!

Over hours spent in the lab or coffee shops, there were four people that I never had a chance to meet but they were always there, helping me be more (or less) productive: Jean Sibelius, deafheaven, Ariana Grande and Kekę. Thank you for your music!

Finally, none of the above would have ever happened if my family hadn't been there for me. Kosma - a true friend for good and worse, thank you for all the cycling trips and many more. Bazyli, Joanna, Marek and Jan for all video calls and support. My Polish Deep State - thank you Marcin, Adam, Marcin, Sławek, Piotr, Michał and Maciek for countless breakfasts and political discussions. In the end, the whole effort was worth it because of my exceptional and wonderful parents. Mamo i Tato, thank you for all your sacrifice and love.

Abstract

Recent developments in machine learning along with a general shift in the public attitude towards digital personal assistants has opened new frontiers for conversational systems. Nevertheless, building data-driven multi-domain conversational agents that act optimally given a dialogue context is an open challenge. The first step towards that goal is developing an efficient way of learning a dialogue policy in new domains. Secondly, it is important to have the ability to collect and utilise human-human conversational data to bootstrap an agent's knowledge. The work presented in this thesis demonstrates how a *neural dialogue manager* fine-tuned with *reinforcement learning* presents a viable approach for learning a dialogue policy efficiently and across many domains.

The thesis starts by introducing a dialogue management module that learns through interactions to act optimally given a current context of a conversation. The current shift towards neural, parameter-rich systems does not fully address the problem of error noise coming from speech recognition or natural language understanding components. A *Bayesian* approach is therefore proposed to learn more robust and effective policy management in direct interactions without any prior data. By putting a distribution over model weights, the learning agent is less prone to overfit to particular dialogue realizations and a more efficient exploration policy can be therefore employed. The results show that deep reinforcement learning performs on par with non-parametric models even in a low data regime while significantly reducing the computational complexity compared with the previous state-of-the-art.

The deployment of a dialogue manager without any pre-training on human conversations is not a viable option from an industry perspective. However, the progress in building statistical systems, particularly dialogue managers, is hindered by the scale of data available. To address this fundamental obstacle, a novel data-collection pipeline entirely based on crowdsourcing without the need for hiring professional annotators is introduced. The validation of the approach results in the collection of the Multi-Domain Wizard-of-Oz dataset (MultiWOZ), a fully labeled collection of human-human written conversations spanning over multiple domains and topics. The proposed dataset creates a set of new benchmarks (belief tracking, policy optimisation, and response generation) significantly raising the complexity of analysed dialogues.

The collected dataset serves as a foundation for a novel reinforcement learning (RL) -based approach for training a multi-domain dialogue manager. A Multi-Action and Slot Dialogue Agent (MASDA) is proposed to combat some limitations: 1) handling complex multi-domain dialogues with multiple *concurrent actions* present in a single turn; and 2) lack of interpretability, which consequently impedes the use of intermediate signals (e.g., dialogue turn annotations) if such signals are available. MASDA explicitly models system acts and slots using intermediate signals, resulting in an improved task-based end-to-end framework. The model can also select concurrent actions in a single turn, thus enriching the representation of the generated responses. The proposed framework allows for RL training of dialogue task completion metrics when dealing with concurrent actions. The results demonstrate the advantages of both 1) handling concurrent actions and 2) exploiting intermediate signals: MASDA outperforms previous end-to-end frameworks while also offering improved scalability.

Table of contents

List of figures xiii

List of tables xvii

1. Introduction 1
- 1.1. Motivation 2
- 1.2. Contributions 3

2. Overview of Spoken Dialogue Systems 7
- 2.1. Architecture of Dialogue Systems 7
- 2.2. Automatic Speech Recognition 7
- 2.3. Domain Ontology and Databases 9
- 2.4. Natural Language Understanding and Dialogue State Tracking 9
- 2.5. Dialogue Act and Dialogue Action Spaces 12
- 2.6. Dialogue Management 14
  - 2.6.1. Dialogue Management as a Reinforcement Learning Problem . . . 15
  - 2.6.2. Markov Decision Process 16
  - 2.6.3. Partially Observable Markov Decision Process 19
  - 2.6.4. Policy Optimisation Methods 20
  - 2.6.5. Learning Policy from Data and Through Direct Interactions 26
  - 2.6.6. Multi-Domain Dialogue Modelling 27
- 2.7. Natural Language Generation 28
- 2.8. Text to Speech Synthesis 29
- 2.9. Modular and End-to-End Architectures 30
- 2.10. Summary 32

3. Efficient Neural Network-Based Dialogue Policy Management 33
- 3.1. Motivation 33
- 3.2. Deep Value-Based Reinforcement Learning 35

3.3. Uncertainty Estimates in Deep Neural Networks 36
  3.3.1. Four Ways to Extract Uncertainties 36
3.4. Evaluation Protocol 39
  3.4.1. Training and Testing Procedures 40
  3.4.2. Comparison with Baselines 42
  3.4.3. Adaptation to Different Environments 43
  3.4.4. Comparison of Computation Complexity 44
  3.4.5. Noise-Robustness In Adversarial Environment 45
3.5. Conclusion 46

4. MultiWOZ - Large Scale Dataset for Multi-Domain Modelling 49
4.1. Motivation 49
4.2. Overview of Data Collections Paradigms 52
  4.2.1. Machine-to-Machine 52
  4.2.2. Human-to-Machine 53
  4.2.3. Human-to-Human 54
4.3. Data Collection Set-up 56
  4.3.1. Dialogue Task 57
  4.3.2. User Side 57
  4.3.3. System Side 59
  4.3.4. Annotation of Dialogue Acts 60
  4.3.5. Data Quality 61
4.4. The MultiWOZ Dialogue Corpus 61
  4.4.1. Data Statistics 61
  4.4.2. Data Structure 63
  4.4.3. Comparison to Other Structured Corpora 63
4.5. A New Wave of Datasets and End-to-End Modelling 64
  4.5.1. New Data Collection Paradigms and Task-Oriented Corpora 64
  4.5.2. Dialogue System Technology Challenge 8 - Multi-Domain Task Completion Track 65
4.6. Conclusion 65

5. Towards End-to-End Multi-Domain Dialogue Modelling 67
5.1. Motivation 67
5.2. Multi-domain Neural Dialogue Manager 69
5.3. Evaluating the state of the art model on MultiWOZ 70
  5.3.1. Intent Modelling 71

- 5.3.2. Decision Making . . . 72
- 5.3.3. Generation . . . 72
- 5.4. Training Protocol . . . 74
  - 5.4.1. Evaluation . . . 74
- 5.5. MultiWOZ as a New Benchmark . . . 75
  - 5.5.1. Dialogue State Tracking . . . 76
  - 5.5.2. Dialogue-Act-to-Text Generation . . . 76
- 5.6. Conclusion . . . 77

6. Multi-Action and Slot Dialogue Agent for Managing Concurrent Actions . . . 79
- 6.1. Motivation . . . 79
- 6.2. Complex Multi-Domain Dialogues . . . 80
- 6.3. Managing Concurrent Actions . . . 82
  - 6.3.1. Explicit Policy Module . . . 82
  - 6.3.2. Sigmoid vs. Softmax . . . 84
  - 6.3.3. Pretraining with Supervised Signal . . . 85
  - 6.3.4. RL-Based Fine-Tuning . . . 86
- 6.4. Multi-Action and Slot Dialogue Agent . . . 86
  - 6.4.1. Joint Modeling of Actions and Slots . . . 87
  - 6.4.2. Training Procedure . . . 88
- 6.5. Experiments and Results . . . 89
  - 6.5.1. Experimental Setup . . . 89
  - 6.5.2. Supervised Learning Phase . . . 90
  - 6.5.3. Multi-Action Reinforcement Learning . . . 90
  - 6.5.4. Evaluating the Full MASDA Model . . . 91
- 6.6. Conclusions . . . 92

7. Conclusion . . . 93
- 7.1. Summary of Contributions . . . 94
- 7.2. Challenges and Future Work . . . 96

References . . . 101

Appendix A. Comparison of Generated Dialogues . . . 123
- A.1. Cam676 Generation Examples . . . 123
- A.2. MultiWOZ Generation Examples . . . 124

Appendix B. Parameters Overview of MASDA . . . 127

List of figures

2.1. An architecture of spoken dialogue systems consists of six main blocks. In the typical modular approach, the components are trained separately. The end-to-end approaches combine internal modules (the grey block). 8

2.2. The dialogue state tracking module carries across turns the key concepts brought up so far in the conversation. The tracking involves both detecting newly mentioned slots and updating the values of the previously mentioned slots. 11

2.3. The dialogue act taxonomy allows a summarization of the key semantics behind the user or system utterances. Several dialogue acts could be used to fully encompass the semantical meaning of more complex utterances. . . . 12

2.4. Markov Decision Process (left) and Partially Observable Markov Decision Process (right). In both cases the next state s_{t+1} is dependent on history of states only through the previous state. 17

2.5. The natural language generation module maps the dialogue acts with slot-value pairs to natural language. 29

3.1. A high-level overview of the deep RL policy network architecture. The input from the tracker state is processed to estimate the Q-value function. Once the action is chosen, the realized dialogue act is combined with information taken from the knowledge base. 41

3.2. The success rate learning curves for BDQN, GPSARSA, DQN, DQN with dropout and DQN with concrete dropout under noise-free conditions for CamRestaurants and SFRestaurants domains, with two standard error bars. . 43

3.3. The success rate learning curves for BDQN, GPSARSA, DQN, DQN with dropout and DQN with concrete dropout with different confusion rates in the CamRestaurants domain. 44

3.4. The success rate learning curves for BDQN, GPSARSA, DQN, DQN with dropout and DQN with concrete dropout with different confusion rates in the SFRestaurants domain. 45
3.5. The training time of GP and BDQN methods in the CamRestaurants domain (left) and in the SFRestaurants domain (right). The time is presented in seconds and averaged over 10 runs. 46
3.6. The success rate learning curves for BDQN, GPSARSA, DQN, DQN with dropout and DQN with concrete dropout in CamRestaurants with different confusion rates during training and a 45% confusion rate at testing. 47
3.7. The success rate learning curves for BDQN, GPSARSA, DQN, DQN with dropout and DQN with concrete dropout in SFRestaurants with different confusion rates during training and a 45% confusion rate at testing. 48

4.1. The hypothetical two turns of a conversation from the customer service domain along with the annotations over the dialogue state, the knowledge base and the user's private information from the database. 53
4.2. The machine-to-machine paradigms generate first template dialogues with a pre-built user simulator and the dialogue system. The sketches of dialogues are further paraphrased with rules or crowd workers. 54
4.3. The Human-to-machine (H2M) paradigm requires a pre-built dialogue system. The annotation of system actions comes for free but the dialogues are biased by the system's capacity and the user's susceptibility to interactions. 55
4.4. The WOZ data collection setup. The user is given a predefined task while the "system" is equipped with knowledge databases to help fulfill the user's goal. All actions on both sides can be tracked to collect a variety of learning signals stored as rich dialogue annotations. 56
4.5. A sample task template spanning over three domains: hotels, restaurants and booking. The sub-goals are gradually introduced during the dialogue. . . . 58
4.6. The interface of the user side. The crowd-worker has to first read the dialogue history. Then, the response box becomes available. The user is also asked to annotate the topics mentioned in the current turn. 58
4.7. The interface of the wizard side. The crowd-worker has to first read the dialogue history. Then, the window with a list of databases activates. The crowd-worker updates the information given by the customer. The lookup button sends the query to the database and provides the list of entities satisfying user requirements. Finally, the response box becomes available. 59

4.8. The annotation set-up for one turn. First, the crowd-worker chooses the domain that the current dialogue concerns. Then, she finishes the suggested sentences annotating dialogue acts with slot and values 60
4.9. Dialogue length distribution (left) and distribution of number of tokens per turn (right). 62
4.10. Dialogue acts frequency (left) and number of dialogue acts per turn (right) in the collected corpus. 62

5.1. An architecture of the multi-domain dialogue system composed of three sources of input: 1) the oracle belief state, 2) the database pointer and 3) the dialogue context. 71

6.1. High-level architecture of the multi-domain management and response generator with three losses. 83
6.2. High-level architecture of the policy module in the proposed MASDA model described in Section 6.4. 87

List of tables

3.1. The list of all slots used in two analyzed domains. The CamRestaurants is a less complex domain with 7 slots. The SFRestaurants uses 11 slots. 40
3.2. The list of available system actions. The slots from Table 3.1 are combined with dialogue act to a form final representation for some actions. 41

4.1. The comparison of number of dialogues in publicly available corpora for modelling task-oriented dialogues before the collection of our corpus. . . . 50
4.2. Full ontology for all domains in our data-set. The upper script indicates which domains it belongs to: *: universal, 1: restaurant, 2: hotel, 3: attraction, 4: taxi, 5: train, 6: hospital, 7: police. 63
4.3. Comparison of our corpus to other largest task-oriented data sets. Numbers in bold indicate best value for the respective metric. The numbers are provided for the training part of data except for FRAMES data-set were such division was not defined. 64

5.1. Performance comparison of different model architectures using a corpus-based evaluation. The results are averaged over three models. 75
5.2. The test set accuracies *overall* and for *joint goals* in the restaurant sub-domain. 76
5.3. The test set slot error rate (SER) and BLEU on the SFX dataset and the MultiWOZ restaurant subset. 77

6.1. The list of all 14 labelled actions occurring in the MultiWOZ corpus. The system specific flag signifies whether the action typically depends on the turker communication style. 82
6.2. The performance of the baseline model and two considered policy modules on the *Constrained* action set (10 actions) relying only on the SL phase. . . 90
6.3. Performance on the *Full* MultiWOZ action set (14 actions); SL phase only. 91
6.4. Results using the RL fine-tuning phase based on the loss function from Equation (6.15). 91

6.5. Results of the full MASDA model which leverages both dialogue act and slot information. 92

A.1. Two dialogues from Cam676 where the system's responses were generated by the model. 123
A.2. First dialogue from MultiWOZ where the system's responses were generated by the model. 124
A.3. A second dialogue from MultiWOZ where the system's responses were generated by the model. 125

B.1. The hyper-parameters set-up for models from Chapter 5 (Baselines) and Chapter 6 (MASDA). 127

Chapter 1

Introduction

In 1950, Alan Turing proposed the *Imitation Game* (Turing, 1950), where he stated:

> "The question and answer method seems to be suitable for introducing almost any one of the fields of human endeavour that we wish to include."

The so-called Turing test re-defines the quest of creating a machine that can think and exhibit human-level intelligence through a conversational perspective. Rather than defining the nature of human consciousness or self-awareness, the thoughtful machine could *equivalently* try to impersonate a real human in open conversation without revealing its identity. Since then Conversational Artificial Intelligence (Conversational AI) has become one of the long-standing challenges in computer science and artificial intelligence fields.

The recent progress in machine learning, particularly in deep neural networks (LeCun et al., 2015; Schmidhuber, 2015), significantly increased the rate of adoption of data-driven systems throughout many industries (Lorenzo-Trueba et al., 2019; Wu et al., 2016). The area of digital conversational assistants might be seen as a prominent example of this rapid change, as Apple's Siri, Google's Assistant or Amazon's Alexa have become daily companions for unprecedented number of people.[1] Although natural language understanding components are powered by deep neural networks trained on a large amount of data and interactions (Henderson et al., 2017; Kim et al., 2018), the reasoning systems behind these assistants are highly handcrafted. Therefore, the market demand led to the creation of modern variants of the Imitation Game like the *Alexa Prize* Challenge (Ram et al., 2018). The challenge requires social bots to converse coherently and engagingly with humans on popular topics such as sports or politics over 20 minutes. It is perhaps striking that many of the proposed systems have found solutions dating back to the early '60s often useful as a system response. These systems use a pattern matching approach with the conversational logic-driven entirely

---

[1] https://www.statista.com/statistics/973815/worldwide-digital-voice-assistant-in-use/

through handcrafted rules. A prominent example of these early models, ELIZA, imitated the language of a psychotherapist with simple word reordering (Weizenbaum et al., 1966). A majority of the challenge participants used similar handcrafted sub-policies as a part of their systems. The above findings suggest that human conversation is inherently complex and ambiguous, and training an open-domain conversational AI that can perform arbitrary tasks is still very far-off (Vinyals and Le, 2015).

As a consequence, instead of focusing on creating ambitious conversational agents that can reach human-level intelligence, the industrial practice has focused on building task-oriented dialogue systems (Young et al., 2013a) that can help with specific tasks, such as flight reservations (Seneff and Polifroni, 2000) or providing bus information (Raux et al., 2005). Such systems operate on well-defined problems where human users interact with a pre-defined goal, and the dialogue assistants merely help with providing a service. Nevertheless, task-oriented spoken dialogue systems (SDS) are still complex, as they have to solve many challenging problems at once under significant uncertainty. For example, spoken language needs to be automatically recognised, the meaning of the utterances has to be decoded, and the user's goal must be comprehended. Further, SDS has to keep track of the history of a conversation, determine what information to convey to the user, convert that information into natural language, and synthesise the sentences into speech that sounds natural. A statistical approach to dialogue modelling has proven to be an effective way of building such conversational agents (Young et al., 2013a).

The above advancements established the necessary technology for building the first generation of commercial SDSs deployable as regular household items. Examples of such systems are Amazon's Alexa, Google's Home or Apple's Siri. Although these systems brought dialogue assistants into our households, they are very limited in terms of the dialogue breadth, providing only basic services. The incorporation of new services and domains is inevitable. However, this raises the question of the scalability of current frameworks.

1.1 Motivation

The central part of dialogue systems concerns dialogue reasoning over the current state of the conversation and taking a useful action for the next conversation's turn (referred commonly as dialogue policy or dialogue management). The most common industrial practice is currently to model the dialogue manager (DM) through hard-coded rules (Pieraccini and Huerta, 2005). This approach requires domain and dialogue experts for the system's conception and maintenance. On the other hand, interaction-based approaches allow for learning the dialogue policy through conversations with human subjects (Gašić et al., 2011) alleviating the need for writing non-trainable logic rules. However, these approaches suffer from scalability issues and a lack of grounding in natural conversations. The main goal of this thesis is to make policy management more efficient and scalable to more complex and long-term conversations over a much larger range of domains.

The first objective is approached by making neural-based approaches more interaction-efficient. The current state-of-the-art Gaussian Process SARSA (GP-SARSA) estimates uncertainties and samples actions that lead to a better user experience but at the expense of a greater computational complexity (Gašić and Young, 2014). Deep reinforcement learning methods are very promising but rely on heuristical exploration, subjecting the user to a random choice of actions during learning (Fatemi et al., 2016; Su et al., 2017). This problem is tackled by applying Bayesian approaches to action exploration. Bayesian uncertainty estimates allow to a guided search targeting under-explored parts of the policy. As a result, the best policy can be found with smaller number of interactions.

The second and main objective is tackled through the reinforcement of data-driven approaches to dialogue modelling. This starts from a critical analysis of the currently available corpora illustrating how under-resourced task-oriented dialogue modelling is, in terms of data. To combat this issue, this thesis proposes a new data-collection paradigm of human-human conversations that led to the collection of the largest task-oriented multi-domain corpus publicly available. The acquired dataset validates the framework and creates a new set of challenges for a variety of dialogue problems. The dataset serves as a foundation for training a neural multi-domain dialogue system. The naturalness of conversations in the corpus shows the necessity of incorporating a richer expressivity of dialogue agent actions. This is achieved by issuing multiple actions at once with new reinforcement learning approaches.

1.2 Contributions

The main topic of this thesis is *multi-domain dialogue policy management*. The thesis is split into two main parts, starting with an overview of the architectures behind spoken dialogue systems. Chapters 2 and 3 cast dialogue policy management as a reinforcement learning problem and present how uncertainty estimates can increase the efficiency of neural network-based models. Then, Chapters 4, 5, and 6 move to data-driven dialogue management. Below, we briefly summarise all subsequent chapters.

Chapter 2 - Overview of Spoken Dialogue Systems. This chapter presents a typical architecture behind spoken dialogue systems. It briefly discusses the main components of the system including automatic speech recognition, natural language understanding, dialogue management and natural language generation. The dialogue policy optimisation problem is discussed extensively. We show how it can be cast as a reinforcement learning paradigm. The literature overview is presented summarising major works in the area with a discussion of their applicability to multi-domain conversations. The chapter concludes with a discussion of the potential of emerging end-to-end approaches.

Chapter 3 - Efficient Neural Network-based Dialogue Policy Management. This chapter shows how uncertainty estimates can improve learning speed in policy optimisation with neural networks. Several Bayesian methods are benchmarked against a state-of-the-art non-parametric approach. The results show that Bayes-by-Backprop (Blundell et al., 2015) consistently provides superior performance being on par with a non-parametric model while greatly reducing computational complexity. Moreover, the explicit modelling of uncertainty estimates improves the robustness of the learnt policy in adversarial environments. This research work has been presented in the publication:

- C. Tegho, P. Budzianowski, M. Gašić , *Benchmarking Uncertainty Estimates with Deep Reinforcement Learning for Dialogue Policy Optimisation*, In Proceedings of ICASSP, 2018. (C. Tegho ran evaluation of different models and write evaluation chapters in the paper, P. Budzianowski proposed the research agenda, developed benchmarked models and wrote a draft of the paper, M. Gašić co-wrote the paper)

Chapter 4 - MultiWOZ - Large-Scale Dataset for Multi-Domain Modelling. The chapter starts by addressing the fundamental challenge of dialogue modelling: a lack of large-scale datasets of task-oriented natural conversations. To address this problem, a data collection procedure is proposed that relies entirely on crowdsourcing without the need for hiring professional annotators. As a result of the approach, the Multi-Domain Wizard-of-Oz dataset (MultiWOZ) is presented – a fully-labelled collection of human-human written conversations spanning over multiple domains and topics. Part of the research work has been presented in the following publications:

- P. Budzianowski, T-H. Wen, B-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, M. Gašić, *MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling*, In Proceedings of EMNLP, 2018.

- O. Ramadan, P. Budzianowski, M. Gašić, Large-Scale Multi-Domain Belief Tracking with Knowledge Sharing, In Proceedings of ACL, 2018. (O. Ramadan proposed a new model, design the model structure and ran evaluation, and co-wrote the paper, P.

Budzianowski prepared the data for model development, oversaw the model inception and benchmarking, and co-wrote the paper, M. Gašić co-wrote the paper)

Chapter 5 - Towards End-to-End Multi-domain Dialogue Modelling. In this chapter, a set of benchmarks is defined – dialogue state tracking, policy optimisation and dialogue-act-to-text generation. The complexity of a new dataset is demonstrated across all three problems. A particular focus is given to building a multi-domain dialogue architecture with a cross-domain database pointer and the attention over the input sequence. The results of this baseline model show the usability of the data and set a baseline for future studies. Part of the research work has been presented in the following publication:

- P. Budzianowski, I. Casanueva, B-H. Tseng, M Gašić. *Towards end-to-end multi-domain dialogue modelling*, Technical Report, University of Cambridge, Engineering Department, 2018.

Chapter 6 - Multi-Action and Slot Dialogue Agent for Managing Concurrent Actions. This chapter presents a Multi-Action and Slot Dialogue Agent (MASDA) model. The model introduces a novel reinforcement learning (RL)-based approach for training multi-domain policies. Building upon the baseline from the previous chapter, MASDA models system acts and slots using intermediate signals, resulting in an improved task-based end-to-end framework. The model can also select concurrent actions in a single turn: this enriches the representation of generated responses. Its effectiveness is demonstrated by achieving state-of-the-art results.

Lastly, **Chapter 7** concludes the thesis and provides a critical summarisation of the proposed methods and models and discusses how the presented research could be incorporated into conversational assistants already deployed. The chapter finishes by outlining potential future research in the light of recent tremendous advances in the area of transfer learning. The conclusions has been guided by the following initial work:

- P. Budzianowski, I. Vulić, *Hello, It's GPT-2 - How Can I Help You? Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems*, The 3rd Workshop on Neural Generation and Translation, In Proceedings of EMNLP, 2019.

Chapter 2

Overview of Spoken Dialogue Systems

This chapter provides a background to the building blocks of spoken dialogue systems. Modular and end-to-end approaches are discussed with a typical taxonomy of task-oriented dialogue systems.

2.1 Architecture of Dialogue Systems

Spoken dialogue considered in this thesis is a simplified proxy of natural spoken conversations. The conversation happens always between only two parties. Typically, one of the sides acts as a petitioner (*user*) with either an implicit or explicit pre-defined goal to achieve. The second side acts as a service (*system*) providing the necessary help for achieving the user's goal. Such goal-oriented dialogue systems are commonly composed of a series of blocks each performing distinct roles (Young et al., 2013b). Figure 2.1 presents an archetypal structure of spoken dialogue systems.

The longstanding research paradigm divides the understanding, reasoning and generation into separated modules. In this pipelined procedure, the waveform of the user's speech forms the input and the system outputs the audio with the answer. This constitutes *one turn* of a conversation. In recent years, the sequence-to-sequence learning approach (Sutskever et al., 2014) pushed the unification of core components (grey block) into one model. A more detailed overview of each component and architecture type is given in the following sections.

2.2 Automatic Speech Recognition

Automatic speech recognition (ASR) is a speech interface transcribing the audio signal into textual form. Early approaches relied on combining speech-dependent heuristics with finding

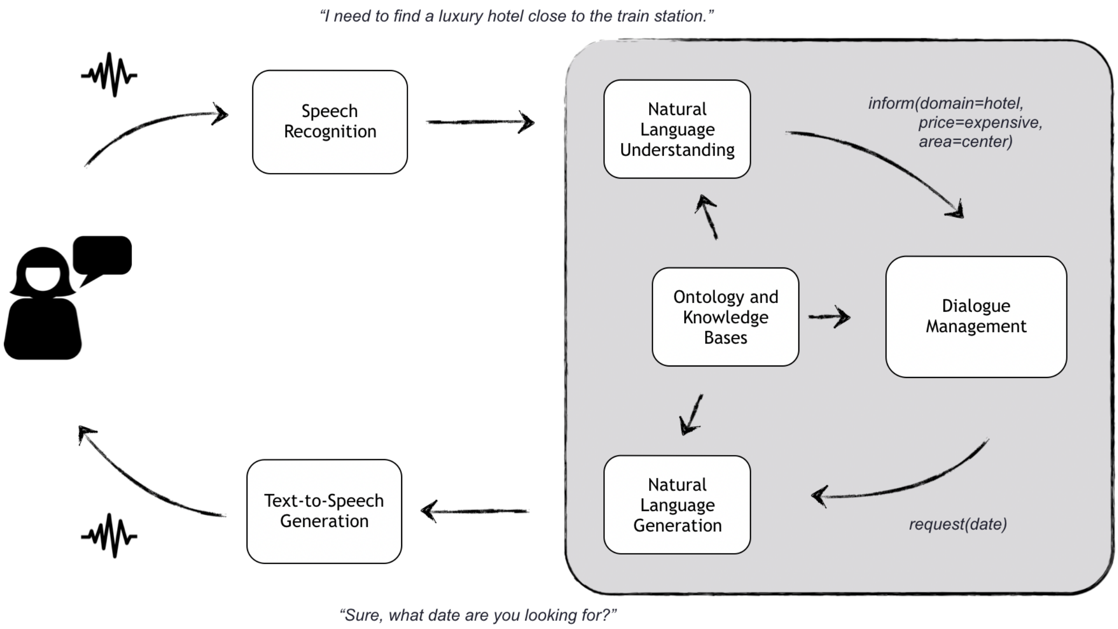

"I need to find a luxury hotel close to the train station."
Speech Recognition
Natural Language Understanding
inform(domain=hotel, price=expensive, area=center)
Ontology and Knowledge Bases
Dialogue Management
Natural Language Generation
request(date)
Text-to-Speech Generation
"Sure, what date are you looking for?"

Fig. 2.1 An architecture of spoken dialogue systems consists of six main blocks. In the typical modular approach, the components are trained separately. The end-to-end approaches combine internal modules (the grey block).

an optimal recognition path over the Markov network. One of the most successful examples was the Harpy system, which could recognize over 1,000 words by the end of the 1980s (Lowerre, 1976). A leap forward in the transcription quality was achieved by applying Hidden Markov Models (HMM) (Baum et al., 1970) to estimate transition probabilities along with Gaussian Mixture models for estimating state probabilities (Gales et al., 2008). Recent breakthroughs using deep architectures replaced acoustic and language parts modelling by feed-forward and recurrent neural networks, respectively (Bengio et al., 2003; Dahl et al., 2011; Hinton et al., 2012). However, the supremacy of pure neural-based models working on the raw signal has been achieved only recently (Chiu et al., 2018; Li et al., 2019; Zeyer et al., 2018).

Although these models achieve human-level (or greater) performance on large vocabulary continuous speech recognition, the more demanding environments still pose a significant challenge (Barker et al., 2018). As of today, state-of-the-art models have substantial problems reaching a level of quality that allows for reliable transcription in open and noisy environments. That is why, the next parts of the dialogue system components often heavily rely on the *n-best list* of transcription hypotheses (Henderson et al., 2014b). These multiple hypotheses provide a more robust estimation of the user's speech, making the system more resistant to noise. The resulting top hypotheses can often have a considerable difference in meaning, especially if sentences are short and are not accompanied by enough context. Therefore, dialogue systems often bias the ASR module depending on the current state of the dialogue. For example, if a system asks for confirmation from the user, the biasing can involve increasing the probability of the expected answer and corresponding keyword, for example: yes or no.

2.3 Domain Ontology and Databases

The focus on a specific task constrains the domain of analyzed conversations. This is why the core components of the task-oriented dialogue system (Figure 2.1, grey block) often operate on a pre-defined set of concepts called an *ontology*. The definition of an ontology is tightly coupled with the knowledge bases used by a system. The entities (for example hotels) in domain knowledge bases have particular properties (like price, size, location, name, phone number). Users *inform* about their preferences by providing *values* for specific slots, called informable slots. They can also request detailed information about a particular entity (for example phone number), which are typically called *requestable* slots (Williams et al., 2013). The ontology defines the list of possible slots and values of these entities, which consequently defines the communication channel between users, systems, and databases.

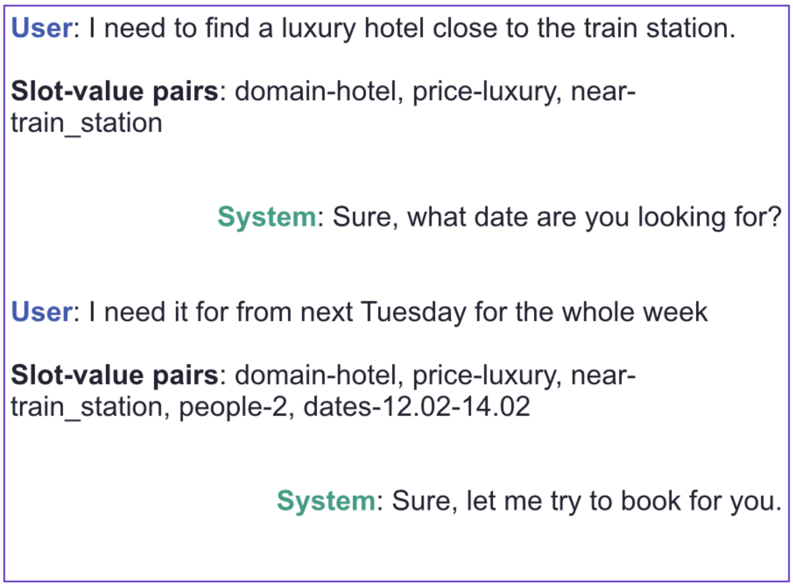
Task-oriented dialogues are tightly coupled with external knowledge bases or databases. The communication channel requires sending discrete API calls for retrieval or direct changes to the database state. For example, a booking procedure requires an update to the database state by sending an API call based on the dialogue state. Further, in many domain-specific applications, the database is an external black-box outside the control of the operating system. This means modelling SDS is not self-contained and prevents training the dialogue policy in an end-to-end fashion. However, some works treat the knowledge base (if it is immutable) as a part of the system (Dhingra et al., 2016; Eric et al., 2017; Madotto et al., 2018).

2.4 Natural Language Understanding and Dialogue State Tracking

The transcribed input from the ASR component is processed in the Natural Language Understanding (NLU) module. NLU seeks to extract the core information from the natural language representation. The fundamental goal of NLU in the case of task-oriented dialogues is the prediction of the right domain and extracting slot-value pairs from the text spoken in a given turn (often called Spoken Language Understanding, SLU (Raymond and Riccardi, 2007)). Originally, systems were based on template-matching (Ward, 1994). Even for relatively small and simple ontologies, the keyword matching approaches fall short when the values are given implicitly. For example, *"I don't care about cost"* does not directly translate to the *expensive* value. This led the way to more data-driven models thanks to the creation of common benchmarks. The Air Travel Information System (ATIS) (Price, 1990) has been commonly used for evaluating SLU models. The first models treated intent determination and inferring slot-value pairs independently (Raymond and Riccardi, 2007; Tur and Deng, 2011). Standard machine learning models were employed based on the Support Vector Machine (Haffner et al., 2003) or Conditional Random Fields (Jeong and Lee, 2008). More recent approaches rely on embedding a sentence through convolutional neural networks (Barahona et al., 2016; Celikyilmaz et al., 2015) or recurrent neural networks (Liu and Lane, 2016; Mesnil et al., 2015). Positive results are being achieved in inferring the semantic meaning directly from audio features without the intermediate text representation (Serdyuk et al., 2018) or merging semantic parsing with deep models (Gupta et al., 2018).

The meaning and information extracted at each turn are useful only if they are considered in the context of the conversation. Carrying the memory across multiple turns is an inherent part of human dialogue. Aggregated information about the progress of such dialogue at each stage of the conversation is called the *dialogue state* (Traum and Hinkelman, 1992). Due to the accumulation of errors coming from transcription and extraction, the dialogue state is represented as a *belief state* with the corresponding distribution of hypotheses over all possible dialogue states. Figure 2.2 presents the updating mechanism of DST.

So far, there have been three lines of approaches to the problem of state tracking (Mrkšić et al., 2017a). The initial systems were based on hand-crafted rules considering only top ASR/NLU hypotheses (Larsson and Traum, 2000; Zhu and Zhao, 2002), with more recent models taking into account top-*n* hypotheses (Wang and Lemon, 2013). Nevertheless, the lack of any learning from observed data and time-consuming parameter tuning for new domains has spurred more data-driven research. The first statistical approaches used deep belief networks to predict the posterior over latent dialogue state given the NLU hypotheses (Thomson and Young, 2010; Williams and Young, 2007). The scalability issues over larger ontologies have been overcome by discriminative approaches directly predicting the conditional probability of the dialogue state given the dialogue history. This includes log-linear models such as the maximum entropy linear model (Lee, 2013) or conditional random fields (Ren et al., 2013). The application of deep recurrent neural networks pushed the performance further achieving state-of-the-art results on recently introduced more challenging corpora (Henderson et al., 2013; Mrkšić et al., 2017a; Perez and Liu, 2017). The dependence on the fixed ontology becomes very restrictive for more complex domains. Wu et al. (2019a)



User: I need to find a luxury hotel close to the train station.

Slot-value pairs: domain-hotel, price-luxury, near-train_station

System: Sure, what date are you looking for?

User: I need it for from next Tuesday for the whole week

Slot-value pairs: domain-hotel, price-luxury, near-train_station, people-2, dates-12.02-14.02

System: Sure, let me try to book for you.

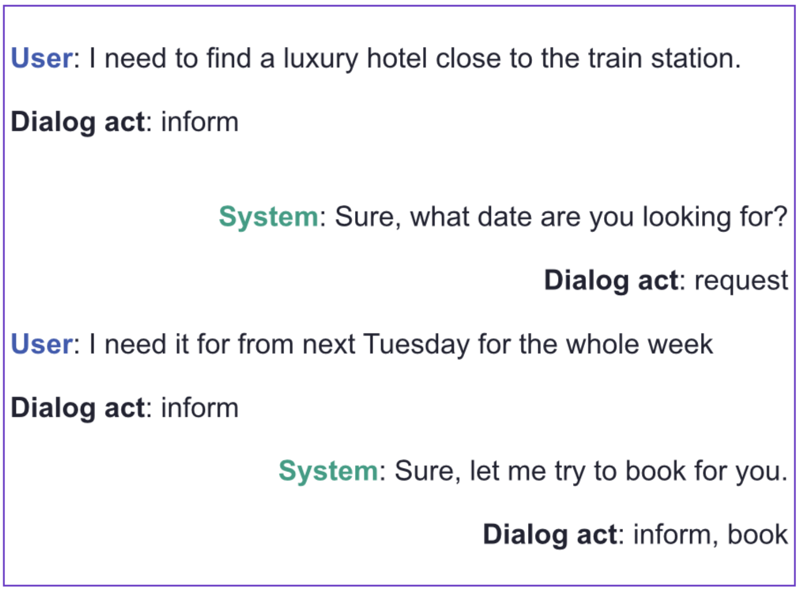
Fig. 2.2 The dialogue state tracking module carries across turns the key concepts brought up so far in the conversation. The tracking involves both detecting newly mentioned slots and updating the values of the previously mentioned slots.

proposed generating dialogue states from utterances using a copy mechanism, facilitating knowledge transfer when predicting (domain, slot, value) triplets not encountered during training. Zhang et al. (2019) merged generation with classification achieving the state-of-the-art results in multi-domain dialogue state tracking. Dialogue state tracking can be also posed as question-answering problem, thus making it naturally extensible to unseen domains, slots, and values (Zhou and Small, 2019).

Throughout this thesis, the true (*ground-truth*) dialogue state will be considered as known for both training and evaluation of presented systems. This means that the true intent(s) and slot-value pairs from user responses are available to the dialogue system. Access to the true dialogue state allows us to also model a variety of scenarios of noise errors propagated either from wrong transcriptions or errors in DST. In turn, it gives a way to test the robustness of the policy logic of the system. We acknowledge that this assumption is not valid when deploying dialogue systems in the wild, but it does offer clear benefits for policy construction and evaluation - the core focus of this work. It facilitates direct comparisons of different policy models by deconflating the mistakes of the policy manager from the errors of preceding modules. x

2.5 Dialogue Act and Dialogue Action Spaces

The ontology defines a domain-constrained vocabulary on which dialogue manager operates. The semantics of this language is encoded through dialogue acts both for user and system replies (Traum, 1999). The semantics typically consists of three parts: dialogue act type and a set of slot-value pairs defined by the ontology. Figure 2.3 shows an excerpt of a conversation with the corresponding dialogue act annotations. The dialogue act type embodies the main action or intent of the sentence and was built upon the speech act concept (Searle, 1969). Although there has been extensive research done on speech act classification in computational linguistics (Stolcke et al., 2000), no automated discovery of speech acts proved superior. In the case of complex utterances, the translation to a semantic layer can be encoded through a set of dialogue acts with corresponding slot-value pairs. The particular subset of dialogue acts that is extensively used in this work, known as the *action space*, consists of actions that the user or the system can respond with. Although it is impossible to list all possible actions, there is a fundamental action set that works as a core set of actions for most task-oriented domains. This set includes:



User: I need to find a luxury hotel close to the train station.

Dialog act: inform

System: Sure, what date are you looking for?

Dialog act: request

User: I need it for from next Tuesday for the whole week

Dialog act: inform

System: Sure, let me try to book for you.

Dialog act: inform, book

Fig. 2.3 The dialogue act taxonomy allows a summarization of the key semantics behind the user or system utterances. Several dialogue acts could be used to fully encompass the semantical meaning of more complex utterances.

1) Request action: The request intent with a particular *slot* signifies that the interlocutor requires more detailed information about the mentioned concept (*slot*). It is a basic dialogue act often used at the beginning of the conversation. The example realization could be: `request(time, date)` - *What date and time would like to book a table?*

2) Confirm action The confirm act allows for resolving uncertainty or reassuring the second party about the current belief state. The uncertainty about the dialogue state often comes from the errors being accumulated along with the conversation. These errors include automatic transcription errors due to noise, wrong predictions of the NLU component, or a misunderstanding of the user or system/user intentions from previous turns. The confirmation action can also serve here as proof of the understanding capabilities for the user when combined with a request action. The example realization is: `confirm(date=26.10.2020, people=4)` - *To recap, we will book you a room on the 26th of October for 4 people. Is that right?.*

3) Select action: If there are multiple entities that satisfy the user's requirements, the system can ask to directly choose one of the options. An example of realizations could be:

```
select(hotel(name=The Great Northern Hotel),
       hotel(name=The Black Lodge))
```

which is equivalent to *There are two hotels in this area - Great Northern Hotel and Black Lodge. Which one would you prefer?*

4) Recommend action: In the case of a large pool of options (for example all hotels in a city), the system can recommend a list of entities to the user that satisfy the requirements already given. The recommendation can consist of one or more entities. For example: `recommend(hotel(name=The Great Northern Hotel))` - *Amongst all the hotels in this area Great Northern Hotels stands out with its reviews.*

5) Inform action: This dialogue act is used to inform the second party about the state of the world or executed internal actions. The inform dialogue act is typically combined with a set of slot-value pairs. In the case of the system, this might be information about the entity that satisfies the client's requirement. For example:
`inform(hotel(name=The Black Lodge,check-in=14:00))` is equivalent to *You can check-in at Black Lodge around 2 pm.* In the case of the user, it might be information requested by the system, for example: `inform(price=expensive)` - *I want something luxurious.* The inform act can be equipped with both slots and values as well it can be overloaded through hidden actions. These actions typically concern sending or receiving API calls to external databases. For example, if payment for the service is required, the system can issue a dialogue act: `inform(payment-completed)` - *We have finalized your transfer* or if the system needs to update the booking slots: `inform(book(hotel(name=The Black Lodge)))` - *The room at Black Lodge has been reserved and your confirmation number is ER293.*

6) Request anything more action: If the main task of the system was achieved, the system can confirm if it should proceed with finishing the conversation by issuing a `request-anything-more` act. An example of natural realization is: *Is there anything more I can do for you?".*

Finally, every dialogue system is equipped with a set of *small talk* actions that enable more human-like interactions such as: `hello` action or `goodbye` action, signifying greetings and the intention to finish the conversation respectively.

2.6 Dialogue Management

The *dialogue manager* (DM) module is a central part of the dialogue system combining the information from the NLU module with the state of knowledge bases to form an action (reply). The action can be either directly realized through the natural language or through system acts that work on the dialogue act level. The former approach belongs to *retrieval-based* systems (Henderson et al., 2017; Lowe et al., 2015). In this work, the latter approach is considered although the action space does not have to be defined explicitly.

The progress in building effective dialogue policies is coupled with better understanding capabilities in the NLU module. Initial approaches modelled the dialogue policy directly by incorporating all possible dialogue realizations through handcrafted rules that create finite-state automata with nodes representing system actions and user responses encoded on edges (Abowd et al., 1995; McTear, 1998). Later on, a method using tree-like structures with decisions at each node corresponding to a finite number of possible user inputs at that particular node was proposed by McTear (2002). Although these policies are easy to design and easy to visualize, they do not scale to more complex domains with many input/output channels. To combat scaling issues, *frame-based* managers were proposed wherein the dialogue state is represented as a frame with slots that need to be filled throughout the dialogue (Kim et al., 2008). However, merely filling appropriate slots cannot model more complex interactions. Traum and Larsson (2003) proposed to use richer (more expressive) representations of the state to perform more sophisticated forms of reasoning. The current state can be modified upon the observation of a new user dialogue move, and the system action is chosen based on the information presented in this updated state. All of these system designs allow for rapid prototyping of dialogue policies with a possibility to visualize dialogue flows. They also guarantee predictable system behaviour that is important from an industry perspective. However, they rely on expert human knowledge about both dialogue management and a specific considered domain. This makes maintenance and scaling to more complex scenarios problematic. Moreover, the noisiness accumulated in the ASR and NLU modules theoretically leads to infinite possible dialogue scenarios even in very restricted domains making modelling through hand-coded rules highly ineffective.

The above limitations led to more data-driven statistical approaches (Levin et al., 2000; Singh et al., 2000b; Young, 2002) that perform automatic optimisation of the dialogue policies either through interactions with the user or through learning from past dialogues. Statistical models provide a framework to incorporate the uncertainty of the dialogue state without writing any logic related to the complexity of the interaction (Lemon and Pietquin, 2007; Williams and Young, 2007). Given the distribution over the input, a probabilistic policy outputs the distribution over policy actions. In a naive approach, the dialogue policy can be seen as a classification task where the dialogue manager needs to choose the right action given the dialogue state. This imitation approach allows for bootstrapping initial policies but cannot achieve planning capabilities. Moreover, the corpus datasets often lack sufficient coverage of all possible situations, and a perfect imitation on the given corpus does not necessarily lead to a successful dialogue interaction (Su et al., 2017).

However, dialogue can be perceived as a sequential decision process optimising a long-term goal, and it can be cast as a control/planning task using reinforcement learning (Sutton and Barto, 1999). Learning dialogue policy through reinforcement learning (*policy optimisation*) defines the conversation as a sequence of actions between two agents with an implicit or explicit reward function. The system aims at learning the policy of actions that will bring the highest reward. In this work, it is associated with helping the user achieve their goal. Originally, the systems were trained assuming a perfect language understanding, leading to casting policy optimisation as a Markov decision process (MDP) (Levin et al., 1998; Singh et al., 2000b). The introduction of multiple hypotheses from NLU led to models that can take advantage of modelling uncertainty in the state. The problem becomes a partially observable Markov decision process (POMDP) (Thomson and Young, 2010; Zhang et al., 2001).

As policy optimisation is the main topic of this work, the dialogue management component is now analyzed in detail. We cast the policy optimisation problem as a partially observable Markov decision problem. The main reinforcement learning strategies applied to dialogue systems are introduced along with an overview of the literature. We discuss the application of data for training dialogue policies and provide an overview of models proposed for tackling complex dialogues spanning several domains.

2.6.1 Dialogue Management as a Reinforcement Learning Problem

Conversations typically involve at least two parties and take multiple turns. Except for small talk, conversations have either implicit or explicit (often long-term) goals. The task-oriented dialogues, specifically, aim at creating value for the user. Both the system and user have goals that they achieve through mutual communication that involve adaptation to the other party through the imperfect channel of natural language. From the perspective of the system, the user is a part of the uncontrollable environment that it interacts with. Early dialogue managers controlled the dialogue flow through a set of hand-crafted rules (Cohen et al., 2004; Mazor and Zeigler, 1995). The *policy trees* depend heavily on the domain and require significant restructuring of the logic if new entities or properties are added to the ontology (Section 2.3). Moreover, those systems had to also take into account inherent uncertainty related to speech-understanding errors which is a challenging task on its own (Section 2.4). Finally, the hand-crafted logic does not capitalize on interactions with the real user and it does not improve over time. The conversational domain is also not a standard classification problem as taken actions directly affect the environment. Furthermore, just following patterns does not directly translate to the goal of dialogue and as such cannot be fully incorporated through a supervised task.

Rather, goal-oriented dialogues fit in a planning framework in which the system learns the optimal way of achieving the goal. Reinforcement learning (RL) (Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1999) is a branch of machine learning that addresses the problem of learning an optimal behavioral strategy. The optimal policy of actions is the one that leads to maximisation of the reward that approximates how likely the dialogue system will help to achieve users' goals. The agent through direct interactions with its environment learns to approximate it either directly or through value function estimation. Dialogue, as such, can be perceived as a sequential decision process in terms of state, action and optimal policy.

2.6.2 Markov Decision Process

Learning through interactions with the environment can be modeled as Markov Decision Process (MDP). For simplicity, first, a discrete-time, finite state and action space case is considered (Puterman, 2014). The MDP can formally be defined as a 5-tuple $(\mathcal{S},\mathcal{A},T,R,\gamma)$. At a given time t an agent observes a world's state s_t that belongs to a finite set of states of the environment $\mathcal{S}$ and executes an action a_t selected from a finite set of actions $\mathcal{A}$. The agent receives a reward r_t which is a realization of a random variable $R(s_{t+1}|s_t,a_t)$ called the reward function $R:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\rightarrow\mathcal{R}$, where $\mathcal{R}$ is a set (possibly continuous) of rewards. Finally, given the chosen action a_t and state s_t, the agent finds itself in the next state s_{t+1} with probability $T:P(s_{t+1}|s_t,a_t)$, where $T:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\rightarrow[0,1]$ is the transition function. The agents main objective is to maximize cumulative discounted reward over (potentially) infinite horizon:

$$R = R(s_{t+1}|s_t,a_t) + \gamma R(s_{t+2}|s_{t+1},a_{t+1}) + \dots \tag{2.1}$$

Here, $\gamma \in [0,1]$ is a discount factor determining how much importance agent gives to the returns of far off future. If $\gamma < 1$, the infinite sum has a finite value as long as the reward sequence is bounded (Sutton and Barto, 1999). The process can be depicted as in Figure 2.4 (left). The MDP satisfies the Markov property if transition probabilities depend solely on the previous state and the action taken, i.e.:

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, ..., s_0, a_0) = P(s_{t+1}|s_t, a_t). \tag{2.2}$$

This property holds in the rest of this chapter.

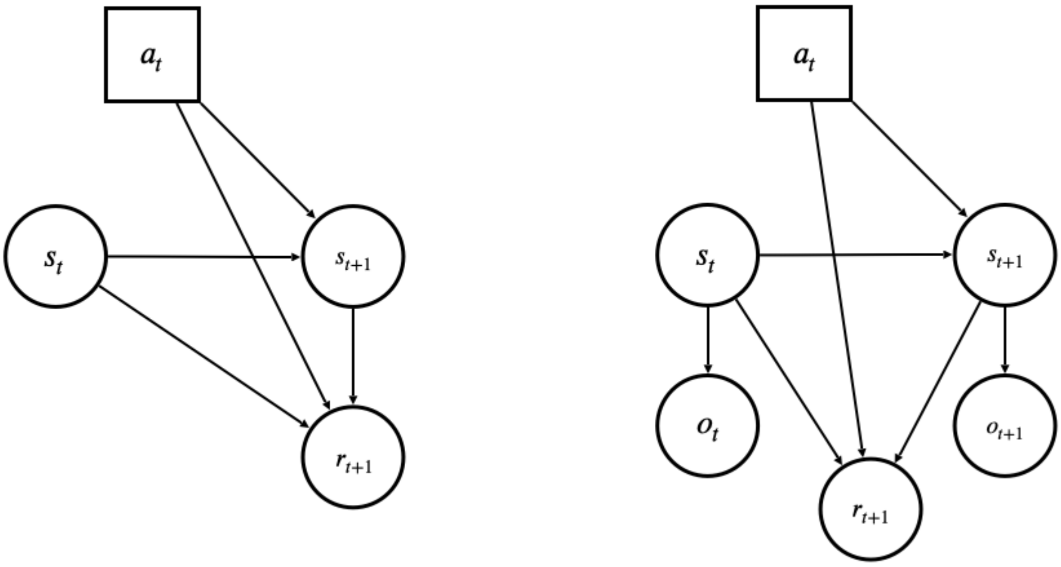


Fig. 2.4 Markov Decision Process (left) and Partially Observable Markov Decision Process (right). In both cases the next state s_{t+1} is dependent on history of states only through the previous state.

The agent takes decision according to the behaviour determined by its *policy*. A policy π is a function $\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$ that with probability $\pi(s,a)$ takes action a in the state s. For any policy π and $s \in \mathcal{S}$, a particular function is of high importance for a family of value-based RL methods - value function V^{π} corresponding to π:

$$V^{\pi}(s) = \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ... \ |s_t = s, \pi\right], \tag{2.3}$$

where $r_t = \mathbb{E}_{a_t \sim \pi(s_t,\cdot)} R(s_{t+1}|s_t,a)$. The goal of optimisation is to find an optimal policy π^*, i.e. a policy that maximizes the value function in each state:

$$V^*(s) = \max_{\pi \in \Pi} V^{\pi}(s). \tag{2.4}$$

Conversely, we might approximate the unique optimal value function V^* which corresponds to any of the optimal policies. Algorithms that seek an optimal policy are called policy search methods while methods approximating value function are called value function estimation algorithms. In next sections we will look at classical algorithms that fall under one of these two categories.

The value function satisfies particular recursive relationships (Bellman, 1954):

$$V^{\pi}(s) = \sum_{s' \in \mathcal{S}} T(s'|s,\pi(s))[R(s'|s,\pi(s)) + \gamma V^{\pi}(s')], \tag{2.5}$$

for s and s' being the current and next state respectively. Equation 2.5 is called the *Bellman equation* for V^{π} and is a base for *dynamic programming* methods. Applying Equation 2.4 we have:

$$V^*(s) = \max_a \sum_{s' \in \mathcal{S}} P(s'|s,a) \left[R(s'|s,a) + \gamma V^*(s')\right], \tag{2.6}$$

known as *Bellman optimality equation*.

The second function of interest is the Q-value function defined as:

$$Q^{\pi}(s,a) = \mathbb{E}\big[R_t + \gamma R_{t+1} + \dots \ |s_t = s, a_t = a, \pi\big], \tag{2.7}$$

and the Bellman optimality equation for Q^* accordingly is:

$$Q^*(s,a) = \sum_{s' \in \mathcal{S}} P(s'|s,a) \left[R(s'|s,a) + \gamma \max_{a'} Q^*(s',a')\right]. \tag{2.8}$$

The optimal policy can now be obtained directly from $Q^*(s,a)$:

$$\pi^*(s) = \arg\max_a Q^*(s,a). \tag{2.9}$$

With the assumption of the known world of $\mathcal{S}, \mathcal{R}, T$ the exact solution to the problem can be obtained by applying recursively the Bellman updates.

Early approaches to dialogues modelled the conversation through the MDP framework. As the environment world is unknown, the empirical model can be built using training dialogue data. Walker (2000) modelled several real systems using simple MDPs, made tractable by reducing the state space to focus on characteristics of specific interest. Singh et al. (2002) proposed building an initial dialogue policy that maps each state to a set of reasonable actions. An initial set of data can be collected following the exploratory behavior of the agent. Given the training data, empirical models of the MDP and reward function are approximate with the optimal policy through 2.8.

2.6.3 Partially Observable Markov Decision Process

The assumption of perfect observability of the world's state is rarely met in practice. This is of central importance for SDS (Section 2.4). However, the problem can be cast within a Partially Observable Markov Decision Process (POMDP) which models an agent acting under uncertainty. The agent can maintain a distribution over all states at each step. This distribution is called the *belief state* and the goal of an agent is learn an optimal policy $\pi(b(s)) \rightarrow a$. The POMDP can be formally defined as a 7-tuple $(\mathcal{S}, \mathcal{A}, T, R, \Omega, \mathcal{O}, \gamma)$ (Kaelbling et al., 1998). At each time step, the agent selects an action a_t moving to unobservable new state s_{t+1} with probability $T(s_t, a_t, s_{t+1})$ and producing a new observation o_t which belongs to the set of observations Ω. The observation probability $\mathcal{O}$ is conditional on a current (unobservable) state s_t and chosen action a_t, i.e. $P(o_t|s_t, a_{t-1})$. The belief at time t is given by the probability distribution over the state s_t, conditioned on the observation o_t, action taken in the previous turn a_{t-1} and the previous belief state, i.e.

$$b(s_t) = P(s_t|o_t, a_{t-1}, b_{t-1}).$$

Figure 2.4 (right) presents graphically the relationship between variables in the POMDP framework.

The optimisation problem in the POMDP framework becomes more challenging than in the case of MDP as the true state is not known explicitly. Nevertheless, the optimal value functions can be computed in a closed form resembling the MDP case (Kaelbling et al., 1998). The optimal value function is defined as:

$$V^*(s) = \max_a \sum_{s' \in \mathcal{S}} P(s'|s,a) \left[R(s'|s,a) + \sum_{o' \in \Omega} P(o'|s,a)\ \gamma V^*(s') \right], \tag{2.10}$$

and the optimal Q-value function is:

$$Q^*(s,a) = \sum_{s' \in \mathcal{S}} P(s'|s,a) \left[R(s'|s,a) + \max_{a'} \sum_{o' \in \Omega} P(o'|s,a)\ \gamma Q^*(s',a') \right]. \tag{2.11}$$

The optimal value function of the belief state can now be derived as:

$$V^*(b(s)) = \sum_{s \in \mathcal{S}} b(s_t = s) V^*(s) \tag{2.12}$$

and similarly the optimal Q-value function as:

$$Q^*(b(s), a) = \sum_{s \in \mathcal{S}} b(s_t = s) Q^*(s, a). \tag{2.13}$$

Policy optimisation for a POMDP, even in the case of known $(\mathcal{O}, T, R)$, is much more complex than for an MDP. In a POMDP, a belief state is a distribution over the underlying state set and is, therefore, a continuous variable. Except for a small set of discrete states, current exact algorithms are effectively computationally intractable. Most scalable approximate methods apply restrictions to value function computations by working on a finite subset of the belief space (Shani et al., 2013).

Early approaches focused on solving simple problems in a simulated environment. In order to avoid the intractability, Roy et al. (2000) proposed ranking belief vectors based on the most likely state and the entropy of the belief state. They found that using a POMDP framework led to better behaviour when recognition errors occurred. Zhang et al. (2001) extended this set-up to a larger ontology comparing the performance of a grid-based POMDP policy optimisation compared to various augmented MDPs. Again, the POMDP version consistently yielded better solutions making POMDP-based frameworks for SDS a promising way forward (Young, 2002).

The slow progress in solving exact POMDPs, even with simplifying assumptions, led research to further approximations. POMDPs can be viewed as a continuous space MDP where every belief is a state. By definition, the resulting MDP has to be considered in the continuous state space since there are infinite beliefs for any given POMDP (Kaelbling et al., 1998). In the following sections, we will assume that we are working in such a belief MDP.

2.6.4 Policy Optimisation Methods

Sections 2.6.2 and 2.6.3 laid the theoretical framework for learning dialogue policy with reinforcement learning. In principle, given the environmental dynamics $(\mathcal{O}, T, R)$ and finiteness of dialogue states $\mathcal{S}$, the optimal policy can be obtained through Bellman recursive updates. In practice, none of these assumptions is true. The dialogue state is not discrete nor do we have access to the transition model of the world. Moreover, the reward function is also not known and it requires further approximations.

However, the progress in function approximation in general machine learning allowed agents to be learned without approximating the environmental dynamics relying entirely on modelling an optimal policy. In general, we can consider two principled ways of learning an optimal dialogue strategy. The *value-based* class of algorithms aims to build a value function that subsequently lets us define a policy. The second family of models, *policy-based* methods, parametrise and learn the policy directly. Both of these families were applied to learning dialogue policy and an overview of these approaches is given in this section.

Value-Function Methods

Monte Carlo Learning. The most heuristic approach to approximating the state-action value function (Equation 2.7) is based on the Monte Carlo framework. As the value of a state-action pair is the expected cumulative future discounted reward, we can estimate it directly from experience. The average of the returns observed after visits to the given state should converge to the expected value by the law of large numbers (Bertsekas and Tsitsiklis, 1996). The first successful application of RL to dialogue management relied on Monte Carlo estimation of the value function (Levin et al., 1998) for learning a dialogue strategy for air travel information system. Nearest neighbor-based MCC model (Lefèvre et al., 2009) extended this model by using a *k*-nearest neighbor scheme. It interpolates the value function and smooths the decision process. In consequence, the model achieves better generalization with learnt policy along with increased robustness to noise coming from the user input.

Fitted Q-learning. The Q-learning family of algorithms (Watkins and Dayan, 1992) approximates the Q-value function using Equation 2.11. The expected immediate reward and the expected maximum action-value of the successor state are replaced by the samples obtained from direct interactions with the environment. The update for the Q-value function at the step k, $Q_k(s,a)$, has the following form:

$$Q_{k+1}(s,a) = (1-\alpha)Q_k(s,a) + \alpha[r + \gamma \max_{a'} Q_k(s',a')], \tag{2.14}$$

where α is a learning-rate parameter that might be time-dependent. The α parameter regulates the speed of training. Under mild conditions $\{Q_k\}$ converges to Q^* with probability 1 for a tabular case with any explorational policy (Bertsekas and Tsitsiklis, 1996).

The direct use of the update 2.14 proved to increase the sample efficiency of learning compared to previous methods (Pietquin and Dutoit, 2006; Scheffler and Young, 2002). However, the increased complexities of analyzed domains required more efficient approaches. Li et al. (2009) employed the least-squares policy iteration algorithm (Lagoudakis and Parr,

2003) where Q-function can be represented as a weighted sum of the features. Although there are no learning parameters to tune the solution scales cubically with the number of features.

Using a deep neural network as a function approximator led to learning of the first successful policies directly from high-dimensional sensory input in challenging game environments (Mnih et al., 2013). The resurgence of the deep learning pushed the advances as well in dialogue management. By applying a deep Q-learning (DQN) model, Cuayáhuitl et al. (2015) showed that deep RL enables learning strategic agents with negotiation abilities. In their study, a deep RL algorithm with a high-dimensional state space was applied with success to strategic board games. The algorithm's robustness relied on the *experience replay* pool which stores update tuples (s, a, r, s') (Lin, 1992). from previous exploration episodes. The size of the pool is often a hyperparameter to optimize. Fatemi et al. (2016) achieved superior performance compared to the state-of-the-art in a noise-free environment on the summary state space. Moreover, Q-learning algorithms do not depend on the current target policy and can be naturally bootstrapped with samples from real-world dialogues (Fatemi et al., 2016; Su et al., 2017).

SARSA. Instead of relying on the experience tuple (s, a, r), $Q(s, a)$ can be approximated by *following* the policy of an agent while learning. This requires to storage of an extended tuple (s, a, r, s', a'). The update rule now becomes:

$$Q_{k+1}(s,a) = (1-\alpha)Q_k(s,a) + \alpha[r + \gamma Q_k(s',a')]. \tag{2.15}$$

Contrary to the Q-learning algorithm which is off-policy, SARSA converges to Q^* with probability 1 only if we consider policy that is greedy with respect to the current action-value function in the limit (Singh et al., 2000a). However, as SARSA performs learning by following the current policy, it is more conservative than Q-learning which is important if training takes place in the real world where the cost of mistakes can be significant.

As in the case of Q-learning, SARSA was employed for dialogue modelling early on without any function approximations (Frampton and Lemon, 2006; Henderson et al., 2008). Learnt policies allow for learning of more effective dialogue strategies on larger input spaces. Moreover, substantial gains in sample efficiency can be achieved when the SARSA update is combined with function approximation. One way to achieve this is to model the Q-function as Gaussian Processes of the form:

$$Q(b,a) \sim \mathcal{GP}(0, k((b,a),(b,a))),$$

where k is the kernel describing correlations in different parts of the space (Engel, 2005). This GP-based Q-function (GPSARSA) is updated by calculating the posterior given the collected belief-action pairs (b,a) and their corresponding rewards (Gašić and Young, 2014). The kernel function k is divided into the kernel for the belief space and the action space. The implicit knowledge of the distance between data points in observation space provided by the kernel made it efficient enough to be trained on a full belief space contrary to previous models (Lefèvre et al., 2009). Moreover, the efficiency of GPSARSA made it possible to train the dialogue policy directly with interaction with real users avoiding the cost of hand-crafting a user simulator (Gašić et al., 2011).

Policy-Based Methods

In policy-based methods, the training objective is to find a parametrised policy $\pi_\theta(a|b)$ that maximizes the expected reward directly:

$$\pi^* = \max_\pi J(\theta) = \max_\pi \mathbb{E}_{s_0,a_0,\dots}\left[\sum_{t=0}^{\infty}\gamma^t R_t\right]. \tag{2.16}$$

The expectation is computed over all possible dialogue trajectories given a starting state s_0. The solution with respect to the policy's parameters θ becomes:

$$\theta^* = \arg\max_\theta J(\theta) = \arg\max_\theta \mathbb{E}_{s_0,a_0,\dots}\left[\sum_{t=0}^{\infty}\gamma^t R_t\right]. \tag{2.17}$$

As in the case of value-based methods, policy-based methods can be divided into model-based and model-free families. Early approaches focused on using model and gradient-free methods to optimize Equation 2.17 through evolutionary strategies (Davis, 1991). Recent works proved that they can often perform on par in robotics (Salimans et al., 2017). Nevertheless, as the sample efficiency is crucial for training dialogue policies through interactions with users, a family of gradient-based methods becomes a more suitable candidate.

Gradient-based learning Denote by $\tau = s_0, a_0, s_1, a_1..$ a trajectory of states while following policy π. Given an initial state s_0 and computing the derivative of the discounted sum of rewards for each trajectory, it follows:

$$\nabla_\theta J(\theta) = \int_\tau \nabla_\theta p(\tau|\pi)R(\tau)\mathrm{d}\tau = \mathbb{E}_{\tau\sim\pi}\left[\nabla_\theta \log p(\tau|\pi)R(\tau)\right]. \tag{2.18}$$

The trajectory probability is dependent on θ only through π which leads to the REINFORCE update (Williams, 1992):

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau\sim\pi(\tau)}\left[(\sum_{t=0}^{T-1} \nabla_{\theta} \log \pi_{\theta}(a_t|b_t))(\sum_{t=0}^{T-1} R(b_{t+1}|a_t, b_t)) \right]. \tag{2.19}$$

Contrary to value-based methods, as the policy is parametrised directly, the distribution over actions is given explicitly. This naturally leads to exploration of policies with higher uncertainty. Moreover, the policy-based optimisation has better convergence properties compared to value-based methods, since the policy is directly modelled and optimised towards the desired objective (Sutton et al., 1999).

Since this form of the gradient has a potentially high variance, a baseline function is typically introduced to reduce the variance whilst not changing the estimated gradient (Sutton and Barto, 1999; Williams, 1992). A natural candidate for this baseline is the value function $V_{\pi_\theta}(s)$. Specifically, the Q-value function can serve as a unbiased estimator of the total reward leading to *policy gradient theorem* (Sutton et al., 1999):

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau\sim\pi(\tau)}\left[\nabla_{\theta} \log \pi_{\theta}(a|b) Q_{\pi_\theta}(b,a)\right]. \tag{2.20}$$

If the Q-value function is modelled separately, most of the approaches in the previous section can be utilised. This can be viewed as a special case of the *actor-critic* architecture, where π_θ is the actor and $Q^{w}_{\pi_\theta}(b,a)$ is the critic, defined by two parameter sets θ and w. The agent (actor) acts by learning the optimal policy while critic evaluates the action by computing the value function.

Early approaches for dialogue modelling with policy-based methods (Jurčíček et al., 2011; Misu et al., 2010; Thomson and Young, 2010) employed the natural actor-critic (NAC) algorithm (Peters and Schaal, 2008). NAC takes the advantage of the Fisher information matrix and it can be viewed as a correction term that makes the natural gradient independent of the parametrisation of the policy and corresponds to steepest ascent towards the objective (Martens, 2014). Fatemi et al. (2016) combined supervised pre-training with RL fine-tuning through interactions analysed showing superior performance compared to state-of-the-art in a noise-free environment. The benchmarks showcased the strong performance of actor-critic models compared to value-based models. More advanced actor-critic methods target the bias and variance problem often observed in empirical studies (Asadi and Williams, 2016; Su et al., 2017; Weisz et al., 2018). Su et al. (2017) employed experience replay with off-policy learning analyzing the effect to sample efficiency. Weisz et al. (2018) experimented with learning in a very large action space, which has two orders of magnitude more actions than previously considered reaching successful policies in many domains. Finally, REINFORCE has been applied with success to dialogue policy learning in an end-to-end framework (Wen et al., 2017a). In this case, the model is pre-trained on human dialogues. The latent action space is discovered using the nearest neighbor approach and the dialogue manager module is finetuned with the REINFORCE algorithm. The finetuned system surpasses the standard modular framework in human tests.

Exploration-Exploitation Dilemma

During training, the agent faces uncertainty with each new action. The fundamental question, particularly important in the early stages of training, is whether to follow the best available policy or risk taking a different action to explore an environment. This problem is often referred to as the exploitation-exploration dilemma (Sutton and Barto, 1999). Selecting an optimal action according to the current policy (exploitation) increases the chance of non-deteriorating performance, however, taking a non-optimal action (exploration) to get more information about the environment may lead to the truly optimal policy. Balancing between both options plays a crucial role in achieving *sample efficiency*: i.e. the minimal amount of steps to learn an optimal policy.[1] The sample efficiency is of crucial importance in real word RL. From an industrial perspective, each interaction of a sub-optimal agent with customers creates additional risks that might translate to financial costs. Therefore, the faster the agent learns a satisfactory policy, the better for both final users as well as the system designer.

The policy-based methods provide an explicit probability distribution over actions. The next action can be sampled according to the distribution naturally guiding the policy based on the uncertainty (Thompson, 1933). The policy gradient methods can also let the algorithm learn by itself how much exploration it needs by explicit entropy regularization (Williams and Peng, 1991). However, value-based methods provide the policy implicitly through maximisation of the value function. In this case, a widely-adopted strategy is the ε-greedy method. This policy selects the optimal action based on the current policy with probability $1-\varepsilon$, or a random action to explore the environment with probability ε. This naive approach introduces natural inefficiency by treating all actions equally. Recently, there has been a rise of interest in incorporating uncertainty estimates with deep architectures (Blundell et al., 2015; Gal and Ghahramani, 2016). These approaches were applied with success also within the deep RL framework (Houthooft et al., 2016; Osband et al., 2016).

---

[1] If the optimal policy is unknown, it is equivalent to achieving a satisfactory level of reward.

2.6.5 Learning Policy from Data and Through Direct Interactions

Throughout this thesis, two training paradigms are considered: 1) learning *from scratch*[2] and 2) bootstrapping the initial policy *from data*. The first paradigm is used when there is no domain-dependent data available and there is easy access to either simulator or human subjects. The second paradigm takes advantage of existing in-domain dialogues. In theory, DM requires both *large corpora* as well as *direct interactions* with real users. In practise none of them is available. The former data source poses high collection costs. Unlike object recognition in computer vision or automatic speech recognition, statistical dialogue modeling is a very ambiguous and not a well-defined problem simply because it lacks consistent annotations and universal well-defined metrics. Annotations in dialogue datasets are therefore also extremely complex. The conversation requires both sides to be active across many turns, and it relies on context: concepts introduced at the beginning of a conversation can be referred to much later. Finally, conversations are also grounded in the real world where visual or audio clues also serve as an integral conversational context. The challenges with the acquisition of the latter learning signal can be explained more trivially. Access to the final user of the system requires the deployment of an actual product. Nevertheless, several efforts were put in employing both sources of signal in training dialogue policies (Gašić et al., 2011; Henderson et al., 2008; Singh et al., 2000b).

Initial approaches to dialogue management employed a small batch of data collected from live use of SDS running a random policy designed to explore the state-action space (Singh et al., 2002, 2000b; Walker, 2000). The data allowed for building empirical MPDs or POMDPs on the state space and computing the optimal dialogue policy. A main limitation of the approach is a forced fixation of the state-space in advance before collecting data as new states (not seen during the collection) can not be added on the fly.

High costs and constraints imposed on the definition of the state-space led to the development of *user simulators* (El Asri et al., 2016; Schatzmann et al., 2006; Scheffler and Young, 2002). In theory, it provides access to an unlimited stream of data and it serves as a great playground for training behaviour policy through interactions entirely from scratch. In practise, since developing a user simulator can be seen as a reverse of creating a dialogue system, substantial simplifications are made in order to make a creation of a simulator possible. A user simulator is typically based on a corpus of human dialogues with agenda-based policy (Schatzmann et al., 2007). First, an initial goal for the user is sampled that might be either highly specific (the user is looking for particular entity) or it is not explicitly defined (the user is looking for a cheap hotel nearby). To introduce more complexity, the goal might not be reachable; for example, the booking dates are not available. The user simulator in

---

[2] It is often denoted as *on-line learning* in the literature (Gašić et al., 2011; Paek and Pieraccini, 2008).

principle should behave rationally and maintain a coherent sequence of actions. In addition, an error model is typically utilised to distort the output in order to simulate noisy conditions (Schatzmann and Young, 2009).

The development of simulators opened the door to training dialogue managers on complex belief states through interactions from scratch (Jurčíček et al., 2011; Li et al., 2009; Scheffler and Young, 2002). Finally, this led to training the dialogue manager using the GP-SARSA algorithm entirely from scratch through direct interactions with human subjects in a narrow restaurant domain (Gašić et al., 2011).

The initial data collections lead to a more rigorous annotation scheme resulting in the creation of a commonly used benchmark in the community – Dialogue State Tracking Challenge (DSTC2) (Williams et al., 2013). The corpus is an example of the human-machine dataset in a restaurant-search domain. Several works used the DSTC2 corpus for bootstrapping in a supervised way through policy-based methods with RL fine-tuning leading to stronger performance on challenging baselines (Asadi and Williams, 2016; Fatemi et al., 2016; Su et al., 2017). However, such dialogues are often sub-optimal preventing the direct application of the RL framework.

The Wizard-of-Oz (WOZ) (Kelley, 1984; Rieser et al., 2005) annotation framework tackles this problem by employing human subjects on both sides of the conversation. Resulting dialogues follow natural human-like interactions providing an "ideal" conversational flow. However, contrary to the human-machine interactions, the annotation signals are not given away directly. Recently, Wen et al. (2017b) proposed a scalable way of collecting turn-level annotations without putting much burden on crowd-workers. This results in collection of several WOZ-style goal-oriented corpora – Cam676 (Wen et al., 2017b), Frames (Asri et al., 2017) and SMD (Eric et al., 2017).

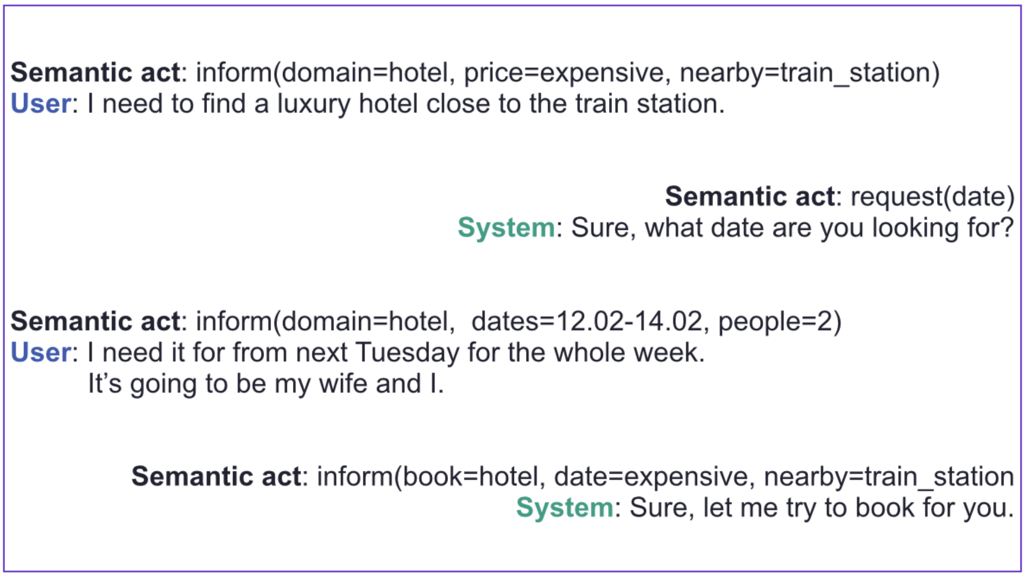
New corpora again sparked research, Wen et al. (2017a) combined the benefits of pre-training dialogue policy with RL finetuning entirely based on the corpus leading to an improved performance against hand-crafted systems. Lei et al. (2018); Wen et al. (2017b) utilise REINFORCE on the generation level to increase the probability of key-words coupled with dialogue success metrics. Nevertheless, the small scale and lack of multi-domain conversations prevent us from pre-training and refining dialogue policies with more parameter-rich models. This will be addressed in more detail in Sections 5.1 and 6.1.

2.6.6 Multi-Domain Dialogue Modelling

All models presented so far targeted single domain conversations leading to highly-constrained frameworks (Ultes et al., 2017). This is in stark opposition to human conversations that often concern several domains, contain cross-references and entailments. However, over the last couple of years, a number of frameworks were proposed for handling multi-domain conversations. The initial multi-domain policies (Cuayáhuitl, 2009; Cuayáhuitl et al., 2010) followed the hierarchical reinforcement learning paradigm (Dietterich, 2000). The agent contains a set of sub-policies that are governed by the meta-policy, each with its own value function. The main limitation of these works comes from the tabular approach which prevents the efficient approximation of state-space crucial for scalability of spoken dialogue systems to more complicated scenarios. Lison (2011) built a policy management algorithm based on the *activation* vector that governs the policy over domains. These vectors inform the hierarchical dialogue manager what policy is in focus at the current state of the dialogue. Wang et al. (2014) shows how central control policy in a master domain can be treated in a sequential planning paradigm rather than a standard classification task. The proposed system uses rule-based policies for sub-domains which can themselves consist of other sub-domains. Gašić et al. (2015) presented a policy committee for adaptation in multi-domain dialogue based on a Bayesian committee machine (BCM) idea. It consists of a number of policies trained on different domains that, at any given time, consult each committee member and chooses the next action using a BCM. Recently, Cuayáhuitl et al. (2016) built a dialogue manager with a set of deep neural networks with application to information-seeking spoken dialogue systems. Although the system operates on multiple domains, the model is not trained using a hierarchical approach as policies for sub-domains are not trained jointly. Instead, the SVM classifier is used to switch between domains that are trained on delexicalised sentences. Next, the chosen DQN is responsible for executing an action in the current turn. This framework allows for transition between domains without following a strict sequence of agents contrary to (Cuayáhuitl, 2009). Budzianowski et al. (2017); Peng et al. (2017) proposed a hierarchical dialogue manager based on the option framework (Parr and Russell, 1998) using deep Q-networks and Gaussian Processes as function approximators.

2.7 Natural Language Generation

The natural language generation (NLG) module takes the policy action in the semantic representation form and transforms it into a natural language that can be understood by users. As in the case of previous modules, the first systems relied on hard-coded *templates* designed for a specific domain. This can be seen as an inverse operation to the one applied in the NLU module. Figure 2.5 shows an example of natural realization of dialogue acts. If system actions are defined explicitly, the template-based generation allows for quick bootstrapping, which provides deterministic realization and robustness.



Semantic act: inform(domain=hotel, price=expensive, nearby=train_station)
User: I need to find a luxury hotel close to the train station.

Semantic act: request(date)
System: Sure, what date are you looking for?

Semantic act: inform(domain=hotel, dates=12.02-14.02, people=2)
User: I need it for from next Tuesday for the whole week.
It's going to be my wife and I.

Semantic act: inform(book=hotel, date=expensive, nearby=train_station
System: Sure, let me try to book for you.

Fig. 2.5 The natural language generation module maps the dialogue acts with slot-value pairs to natural language.

However, the amount of rules typically does not scale linearly with the size and complexity of the domain. Moreover, templates require adaptation to the context of the current dialogue: the same system act should have a different realization with different dialogue states. Finally, by definition, they lack any variability which decreases human interaction satisfaction (Stent et al., 2005). The first attempts at data-driven language generators used sentence planning using rules learnt from the corpus (Walker et al., 2001) and applying boosting algorithms for a better sentence re-ranking (Stent et al., 2004). Another way is to train language models conditioned on the semantic representations using n-gram (Oh and Rudnicky, 2000) or phrase-based models (Mairesse and Young, 2014) and more recently through deep recurrent neural networks (Dušek and Jurcıcek, 2016; Wen et al., 2015, 2016). Juraska et al. (2018) proposed an ensemble of three different encoders (exploiting both recurrency and convolution), reaching state-of-the-art levels on common NLG benchmarks (Dušek et al., 2020)

2.8 Text to Speech Synthesis

The text to speech synthesis (TTS) module is a final part of SDS, transforming the natural text or symbolic representation into audio waves mimicking human speech. The first speech synthesis systems concatenated pre-recorded speech fragments from a human corpus (Hunt and Black, 1996; Moulines and Charpentier, 1990). The limitations of this method, mainly artificial and discontinuous sound due to different recordings, led to research on data-driven methods. The TTS witnessed similar progress, as was the case in ASR.

The first generation of parametric models was based on Hidden Markov Models (Yoshimura, 2002; Zen et al., 2009) that were later superseded by approaches based on deep neural networks (Ling et al., 2015; Ze et al., 2013). The application of autoregressive deep neural networks on raw audio signal pushed the state-of-the-art even further with models such as WaveNet (Oord et al., 2016), thereby achieving human-level quality in some closed domains. Recent models allow for biased speech generation towards specific characteristics of the voice, like gender or age, across many speakers (Ping et al., 2017).

TTS is often considered as an off-the-shelf module in SDS without a direct impact on the general reception of conversational systems. However, the human perception of system intelligence is greatly biased by purely speech synthesis-based characteristics (Cohn et al., 2019).

2.9 Modular and End-to-End Architectures

So far, the components discussed in this chapter have been treated separately, and this has been prominent industrial practice (Pieraccini and Huerta, 2005; Ponnusamy et al., 2019; Shum et al., 2018). However, the lack of knowledge sharing between components introduces many limitations. First, the lack of connection between modules requires to define how the user satisfaction score from interacting with a dialogue system should be translated into distinct components. Secondly, re-training one module affects the whole processing pipeline as the output distribution is the artefact involved in training dependent modules. For example, when one module (e.g. NLU) is retrained with new data, all the others (e.g DM) that depend on it become sub-optimal because they were trained on the output distributions of the older version of the module. The adjustment requires re-training the dependent modules without the assurance of maintaining at least a similar performance. Finally, distinct modules require different data annotation schemes. That led to many resources targeting similar domains but collected in different environments. These resources cannot be shared across modules, thereby reducing the amount of available data. The above-mentioned issues are targeted through modelling some dialogue modules (Figure 2.1, grey block) together. Originally employed for open-domain conversations (Vinyals and Le, 2015), such models often use structures that internally resemble modular systems (Li et al., 2017b; Wen et al., 2017b; Williams and Zweig, 2016). Three lines of architecture simplifications were advocated: 1) training NLU and dialogue management jointly, 2) combining policy design with generation and 3) merging all three components together following open-domain conversational models.

Merging NLU and DM The first family of models aims at merging understanding and reasoning by jointly working on the natural language input. Williams and Zweig (2016) employed a recurrent structure to update the dialogue state across turns and predict the next action. The hidden recurrent state consists of entity extraction, the results of API calls, and previous actions. Further, a hybrid approach taken by Williams et al. (2017) allows for the incorporation of business rules and other prior knowledge via software and action templates while still inferring a latent representation of the dialogue state. Zhao and Eskenazi (2016) avoided the reliance on entity extractors and incorporated state tracking labels directly into the loss. This greatly helps to improve the learning efficiency on a question-game dialogue simulator. Yang et al. (2017) introduced an end-to-end deep recurrent network with a contextual dialog memory that can be jointly trained by three supervised signals: user slot tagging, intent prediction and system action prediction. Liu et al. (2018) merged NLU with DM to tackle issues with mismatching the dialogue state distribution between offline training and online interactive learning stages. The dialogue policy is interchangeably trained through dialogue sampling with human interactions and re-training on collected conversations.

Combining DM and NLG The combination of reasoning and natural language generation was shown to yield performance improvements even in the case of state-based estimation (Lemon, 2011). Wen et al. (2017b) modeled these two components together through a policy network (a feed-forward network) that conditions the initial state of the natural language generator (a recurrent network) that directly outputs words. Their evaluation with human subjects shows a strong preference for the proposed model against the traditional modular architecture. Further, Wen et al. (2017a) added an explicit action layer that allow interpretion of the lower-level implicit system actions before the generation part. The network could be pre-trained with action labels inferred in an unsupervised way. The user-system turn pairs can be clustered together with the number of clusters becoming the size of the action space. The previous two models relied on pre-trained dialogue state trackers. Lei et al. (2018) advocated re-using the recurrent structure for both dialogue state tracking and language generation. The simplification of the architecture reduces the number of parameters and computational complexity while improving the final performance. A more thorough analysis of this family of models is provided in Chapter 5.

Blending NLU, DM, and NLG Together Finally, several architectures put all three core components together without explicit low-level signals of state tracking, dialogue actions or natural language generation. Bordes et al. (2017) showed that an end-to-end dialog system based on Memory Networks (Weston et al., 2014) can reach promising learning levels in performing non-trivial operations: issuing and updating API calls, providing information, and learning basic latent dialogue state tracking mechanisms. The architectures analyzed so far rely on external databases and issue a symbolic query to the knowledge bases to retrieve entries based on their attributes. However, such symbolic operations break the differentiability of the system and prevent end-to-end training of neural dialogue agents. Recent works have proposed incorporating a knowledge base with the dialogue reasoning enabling the propagation of a learning signal throughout the whole network (Dhingra et al., 2016; Eric et al., 2017). Madotto et al. (2018) combined a memory network with the pointer mechanism (Vinyals et al., 2015). The multi-hop attention mechanism allows incorporating a knowledge base and control each generation step.

Experiments in various domains have showed that the above-mentioned models can outperform the state-of-the-art pipeline models in task-completion metrics indicating the potential of end-to-end approaches. Some progress has been achieved with transferring the end-to-end structures across domains in a zero-shot learning manner (Zhao and Eskenazi, 2018). In such a scenario, the dialogue system generalizes to a new domain where only a domain description is provided, and no training dialogues are available.

2.10 Summary

In this chapter, based on the prior work, the typical architecture behind spoken dialogue systems was described and dialogue management was cast in the reinforcement learning paradigm. Two families of algorithms for policy learning were analyzed with a focus on the exploitation-exploration dilemma. We discussed how real interactions can be used to initialize the policy. The overview of dialogue models for more complex, multi-domain dialogues was presented. The current shift towards neural, parameter-rich systems needs to accommodate for several challenges posed by multi-domain conversations. The state-of-the-art value-based algorithms for dialogue management rely on uncertainty estimates that have to be incorporated into neural networks. Moreover, the deployment of a dialogue manager without any pre-training on human conversations is not a viable option from an industry perspective. Finally, the neural-based research on combining multiple components of dialogue systems shows promise for scaling across many domains re-using data across modules. These outstanding issues will be explored further in the next chapters of this thesis.

Chapter 3

Efficient Neural Network-Based Dialogue Policy Management

This chapter aims at advancing the speed and robustness of the dialogue policy learning through reinforcement learning with Bayesian deep neural networks. The first part of the chapter gives an overview of leading methods to obtain approximate estimates of uncertainty with deep neural networks. In the second part, all these methods benchmark against a noisy and adversarial environment of spoken dialogue systems. The results show that Bayesian deep RL can lead to on par results with the state-of-the-art non-parametric models while greatly improving computational efficiency. What is more, the benefits are larger the more adversarial environment becomes, exceeding the performance of the non-parametric model.

The results from sections 3.4 have been joint work with Christopher Tegho as a part of his Master's project and they were published in (Tegho et al., 2018).

3.1 Motivation

Policy management optimisation through reinforcement learning has been applied to more and more challenging domains and environments (Gašić et al., 2011; Weisz et al., 2018; Young et al., 2013b). Recently, deep reinforcement learning (Mnih et al., 2013) proving its scalability and flexibility has been investigated as a continuous learning framework (Asadi and Williams, 2016; Fatemi et al., 2016; Su et al., 2017). However, deep reinforcement learning is known to be highly sample inefficient (Salimans et al., 2017). This is in stark contrast to the requirements of dialogue systems where human interactions are greatly limited.

The key factors improving the efficiency lies in better exploration and generalization to unknown parts of state space (Osband et al., 2016). To explore the environment, an ε-

greedy policy is often employed, where a greedy action is taken with respect to the estimated Q-function with probability $1-\varepsilon$, and a random one with probability ε (Sutton and Barto, 1999). However, given the large action-state space, a randomly exploring agent is not sample efficient. Convergence to an optimal policy is slow and successful dialogues are hard to achieve. This is especially the case in on-line learning, where the user is potentially subjected to poor behaviour.

The Q-function value of each state-action pair can be augmented with an estimate of its uncertainty to guide exploration, and achieve faster learning and a higher reward during learning (Daubigney et al., 2011). The exploration-exploitation dilemma (Section 2.6.4) can be answered by following the current distribution. By sampling from the distribution of the Q-value function, actions with higher variance are more likely to produce extreme (positive and negative) Q-values. These actions are more likely to be chosen and the outcome of these actions will reduce the uncertainty leading to optimal values and in turn to the optimal policy (Thompson, 1933). Gaussian Processes (GPs) provide an explicit estimate of uncertainty (Gašić et al., 2010), overcoming the problem of random exploration. However, GP-SARSA requires computing the inverse of a Gram matrix for determining the predictive posterior and estimating Q at new locations that is dependent on number of training data points. In the case of a dialogue system, this will be equal to the total number of turns, summed overall dialogues, which poses a serious computational problem. Sparcification methods are then needed to achieve lower computational complexity (Casanueva et al., 2015). This may limit however the accuracy of the final solution.

Deep neural network (DNN) models on the other hand scale well with data and are computationally more efficient than GPs. They proved to be well suited for the policy management task (Cuayáhuitl et al., 2015; Fatemi et al., 2016; Su et al., 2017). However, they do not directly provide an estimate of uncertainty, relying on ε-greedy exploration, which lowers the sample efficiency. Building upon recent advancements in Bayesian deep learning (Blundell et al., 2015; Gal et al., 2017; Li and Gal, 2017; Osband et al., 2016), the primary motivation of the work presented in this chapter is to perform an extensive benchmark of uncertainty estimates.

Recent works on employing Bayesian DQN to dialogue management focus on small ontologies and without modelling the errors coming from ASR or NLU modules (Chen et al., 2017; Lipton et al., 2018). Moreover, the results are not compared with the state-of-the-art non-parametric model, GP-SARSA. Therefore, the results do not provide a clear signal of evaluation. Lipton et al. (2018) applied the Bayes-by-Backprop method (BDQN) in a movie domain comparing with the standard DQN model. The results suggest that Bayesian exploration helps to achieve higher success rates than standard exploration. Chen et al.

(2017) shows how uncertainty estimates obtained with dropout can improve the safety and efficiency of policy optimisation. The authors proposed a student-teacher architecture where a data-driven student policy decides when to update its own policy consulting a rule-based system based on uncertainty estimates. Again, the analysed domains are of limited scale with no error modelling in NLU.

Following (Lipton et al., 2018), the Bayes-by-Backprop method with other Bayesian deep RL methods for deep-Q-networks are compared to GP-SARSA. We investigate how BDQN can be improved and be made competitive with GP-SARSA. BDQN learns dialogue policies with more efficient exploration than other deep Bayesian methods, and reaches performance comparable to the state-of-the-art in policy optimisation, namely GP-SARSA, particularly in high noise conditions. We also implement α-divergences, variational dropout, and minimizing the negative log-likelihood as other means to extract uncertainty estimates from DQN, and compare performance to BDQN and DQN.

3.2 Deep Value-Based Reinforcement Learning

The approximation to Q-value function (Equation 2.7) with deep neural networks (Deep Q-Networks, DQN) reached the human level performance on challenging game environments (Mnih et al., 2015). DQN adapts the Q-learning update (Equation 2.14) by iteratively improving the guess of $\hat{Q}$ through minimization of the difference between the target next-step value and the current estimate:

$$L(w_t) = \mathbb{E}_\pi \left[(y_t - \hat{Q}^\pi(b_t, a_t; w_t))^2 \right], \tag{3.1}$$

where the expectation is taken with respect to ε-greedy policy (Mnih et al., 2015). The targets follows from the 2.14:

$$y_t = r_t + \gamma \max_{a'} \hat{Q}^\pi(b_{t+1}, a'; w_t^-) \tag{3.2}$$

with w and w^- being two set of parameters; one for the current step t and one for the target evaluation y_t. The target network w^- is updated less frequently than the network w to stabilise learning. Initially, Q-learning update 2.14 was used to perform an update of the parameters once an episode of exploration was performed. No memory of those episodes was retained limiting the sample efficiency. To further stabilise and improve data efficiency during the training, the experience replay pool (Section 2.6.4) can be employed (Lin, 1992). The pool can be used during an update step to re-use previous episodes.

DQN often suffers from over-estimation of Q-values as the max operator is used to select action as well as to evaluate it. Double DQN (DDQN) decouples the action selection and

Q-value estimation achieving better performance (Van Hasselt et al., 2016). We can untangle the selection and evaluation in Q-learning by rewriting the target:

$$y_t = r_t + \gamma \hat{Q}^{\pi}(b_{t+1}, \arg\max_a \hat{Q}^{\pi}(b_{t+1}, a, w_t); w_t^-). \tag{3.3}$$

In this case, the set of current weights w_t are used to choose the greedy action while to fairly evaluate the value of this policy, we use the target weights w_t^-. The double DQN network with experience replay will be used as a base for further experiments in this chapter.

3.3 Uncertainty Estimates in Deep Neural Networks

Standard neural network provides only point-based estimates of parameters. Bayesian neural networks (BNNs) gives a way to obtain uncertainty estimates from a neural network (Neal, 2012). Instead of having single fixed value weights in the neural networks w, all weights can be represented by probability distributions over possible values given observed dialogues $\mathcal{D}$, $P(w|\mathcal{D})$. In turn, uncertainty in the hidden units allows the expression of uncertainty about predictions (Blundell et al., 2015).

For exploration, Thompson sampling is used instead of ε-greedy, which consists of performing a single stochastic forward pass through the network every time an action needs to be taken. The posterior distribution over the weights given the training data, $P(w|\mathcal{D})$, is calculated with Bayesian inference. The Q-values given the input belief state b are given by:

$$Q(b,a) = \mathbb{E}_{P(w|\mathcal{D})}[Q|b,a,w]. \tag{3.4}$$

Taking an expectation under the posterior distribution is equivalent to using an ensemble of an uncountably infinite number of neural networks, which is intractable (Blundell et al., 2015). We have to resort to sampling-based variational inference or stochastic variational inference.

3.3.1 Four Ways to Extract Uncertainties

Since the use of uncertainty estimates in deep RL is under-explored area (Gal and Ghahramani, 2016; Osband et al., 2016), we analyse four algorithms to extract uncertainty estimates from deep Q-Networks. At the same time, it will allows us to test whether deep RL methods can achieve the sample efficiency comparable to the state of the art non-parameteric algorithm GPSARSA in the dialogue domain. All four methods can be cast within the variational inference framework Hinton and Van Camp (1993).

Variational inference. The intractable posterior $P(w|\mathcal{D})$ is approximated with a variational distribution $q(w|\theta)$. The parameters are learnt by minimizing the Kullback-Lieber ($\mathcal{KL}$) divergence between the variational approximation $q(w|\theta)$ and the true posterior on the weights $P(w|\mathcal{D})$. We therefore look for a set of parameters θ that minimizes:

$$\begin{aligned}
\theta^* &= \arg\min_{\theta} \mathcal{KL}[q(w|\theta)||P(w|\mathcal{D})] && (3.5)\\
&= \arg\min_{\theta} \int q(w|\theta) \log \frac{q(w|\theta)}{P(w)P(\mathcal{D}|w)} \mathrm{d}w && (3.6)\\
&= \arg\min_{\theta} \mathcal{KL}[q(w|\theta)||P(w)] - \mathbb{E}_{q(w|\theta)}[\ln P(\mathcal{D}|w)]. && (3.7)
\end{aligned}$$

The resulting cost function is termed as the variational free energy (Hinton and Van Camp, 1993):

$$\mathcal{F} = \mathcal{KL}[q(w|\theta)||P(w)] - \mathbb{E}_{q(w|\theta)}[\ln P(\mathcal{D}|w)]. \tag{3.8}$$

Deep BDQN-Learning. The Bayes-by-backprop method (Blundell et al., 2015) proposes to optimize 3.8 by sampling from $q(w\theta)$ using the reparameterization trick (Kingma and Welling, 2014). This results in unbiased estimates of gradients of the cost in 3.8 used to learn a distribution over the weights of a neural network. We choose $q(w|\theta)$ to be a Gaussian with diagonal covariance with a variational parameter set θ following previous results (Blundell et al., 2015; Lipton et al., 2018). Given the mean μ_i and covariance σ_i of q for each weight, a sample from q is obtained by first sampling $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon)$, then computing $w_i = \mu_i + \sigma_i \circ \varepsilon_i$, where $\circ$ is point-wise multiplication. To ensure all σ_i are strictly positive, the softplus function $\sigma_i = \log(1+\exp(\rho_i))$ is used where ρ is a free parameter (Lipton et al., 2018). The variational parameters are then $\theta = \{\mu_i, \rho_i\}_{i=1}^{\mathcal{D}}$ for D-dimensional weight vector w. The resulting gradient estimator of the variational objective is unbiased and has a lower variance. The exact cost in Eq. 3.8 can then be approximated as:

$$\mathcal{F}(\mathcal{D}, \theta) \approx \sum_{i=1}^{n} \log q(w^{(i)}|\theta) - \log P(w^{(i)}) - \log p(\mathcal{D}|w^{(i)}) \tag{3.9}$$

where $w^{(i)}$ is the ith Monte Carlo sample drawn from the variational posterior $q(w^{(i)}|\theta)$. For the objective function in Eq. 3.8, we use the expected square loss.

α-Divergences. The approximate inference technique described in the Bayes-by-backprop method corresponds to Variational Bayes (VB), which is a particular case of α-divergence, where $\alpha \to 0$ (Hernández-Lobato et al., 2016). The α-divergence measures the similarity between two distributions and can take the form:

$$D_\alpha[p||q] = \frac{1}{\alpha(\alpha-1)}(1 - \int p(\theta)^\alpha q(\theta)^{1-\alpha} d\theta), \tag{3.10}$$

where $\alpha \geqslant 0$.

Hernández-Lobato et al. (2016) found that using $\alpha \neq 0$ performs better than the VB case, where an approximation with $\alpha \geq 1$ will cover all the modes of the true distribution, and the VB case only fits a local mode, assuming the true posterior is multi-modal (Hernández-Lobato et al., 2016). $\alpha = 0.5$ achieves a balance between the two and has shown to perform best when applied to regression or classification tasks.

We experiment with an objective function based on the black box α-divergence (BB- α) energy. We use the reparametrization proposed by (Li and Gal, 2017):

$$\mathcal{L}_\alpha \approx \check{\mathcal{L}}_\alpha = \mathcal{KL}[q(w|\theta)||P(w)] - \frac{1}{\alpha}\sum_n \log \mathop{\mathbb{E}}_{q(w|\theta)}[P(\mathcal{D}|w)], \tag{3.11}$$

where $\mathcal{L}_\alpha$ designates the BB- α energy, $\check{\mathcal{L}}_\alpha$ designates an approximation, and n corresponds to the number of datapoints in the minibatch.

DQN-MC Dropout. Another method to obtain uncertainty estimates in deep neural networks is Bayesian inference with dropout (Gal and Ghahramani, 2016). Dropout consists of randomly dropping units (with some probability d) from the neural network during training (Srivastava et al., 2014).

As in the previous methods, dropout can be analysed from the variational inference perspective (Equation 3.8). This comes from the fact that applying a stochastic mask is equivalent to multiplying the weight matrix in a given layer by some random noise. The resulting stochastic weight matrix can be seen as draws from the approximate posterior over weights, replacing the deterministic weight matrix (Gal and Ghahramani, 2016).

The prediction uncertainty is obtained by marginalizing over the approximate posterior using Monte Carlo integration (Li and Gal, 2017):

$$\begin{aligned} p(y=c|x,X,Y) &= \int p(y=c|x,w)p(w|X,Y)dw && (3.12)\\ &\approx \int p(y=c|x,w)q_\theta(w)dw && (3.13)\\ &\approx \frac{1}{K}\sum_{k=1}^{K} p(y=c|x,\hat{w}_k) && (3.14) \end{aligned}$$

with $\hat{w}_k \sim q_\theta(w)$. Dropout is a simple technique of variational approximation, without the need to make major changes to the DQN implementation. Dropout variational inference is implemented by adding dropout layers before every weight layer in the neural network model. No changes to the objective function are necessary.

DQN-Concrete Dropout. To obtain well-calibrated uncertainty estimates with the above method, a grid-search over the dropout probabilities is necessary. However, we can treat a dropout as a part of optimisation task obtaining an automatic method of tuning the mask. One method is to continuously relax the dropout's discrete masks and optimize the dropout probability using gradient methods (Gal et al., 2017). Dropout d probability becomes one of the optimized parameters. The concrete distribution relaxation z of the Bernoulli random variable becomes:

$$z = \text{sigmoid}(\frac{1}{t}(\log d - \log(1-d) + \log u - \log(1-u))$$

with some temperature t which results in values in the interval $[0,1]$ and $u \sim \mathcal{U}(0,1)$.

3.4 Evaluation Protocol

Experiments are conducted using the Cambridge and San Francisco restaurant domains (CamRestaurants and SFRestaurants respectively). In both domains, users converse with the system to find restaurants in a city that match their constraints such as food type and area. The Cambridge restaurant domain consists of approximately 100 venues and each venue has six attributes (slots) of which three (area, price range, and food type) can be used by the system to constrain the search. The remaining three (phone number, address, and postcode) are informable properties that can be queried by the user once a database entity has been found. The San Francisco domain is a larger domain with 250 venues and a richer ontology consisting of 10 slots. In this domain, the user might have constraints regarding the decoration type of the restaurant or the exact price of dishes. The additional slots allow to model more complicated dialogue trajectories. Table 3.1 shows the full list of slots for both ontologies.

The conversations are modelled through the PyDial toolkit (Ultes et al., 2017). The toolkit provides the framework for training and evaluating task-oriented dialogues following the modular architecture (Figure 2.1). Conversations could be modelled on the speech, natural text or semantic output levels and the experiments will be run in the latter scenario. The semantic level works entirely on the dialogue act level (Section 2.4) and the provided ontology.

Slots	**CamRestaurant**	**SFRestaurants**
name	✓	✓
address	✓	✓
phone number	✓	✓
postcode	✓	✓
near		✓
area	✓	✓
food type	✓	✓
pricerange	✓	✓
price		✓
decoration type		✓
family welcome		✓

Table 3.1 The list of all slots used in two analyzed domains. The CamRestaurants is a less complex domain with 7 slots. The SFRestaurants uses 11 slots.

It allows to separate the error signals of the dialogue manager by explicitly controlling what is the input quality to the manager. To effectively encode the dialogue history and user goal, the agenda-based user simulator with parameters estimated from data is used (Schatzmann et al., 2007). In this method, the user goal is decomposed into a set of slot-value pairs representing the requests and constraints stored in a stack.

The input to the dialogue manager is a real-valued belief state b_t formed by concatenating different information signals. This includes full belief distribution over the ontology, previous system action, flags representing whether the system requested information about some slots, whether the system has already proposed some entity following Casanueva et al. (2017); Gašić et al. (2011). The belief state size is 256 and 636 in CamRestaurants and SFRestaurants respectively. The system in both domains operates on district action space dependent on the ontology. The summary list of system actions is defined in Table 3.2. Overall, the action space sizes are 14 and 25 in CamRestaurants and SFRestaurants respectively.

The dialogue is considered successful if two requirements hold: 1) the system found an entity that satisfies the constraints given by the user 2) all requested information about the provided entity was given. Additionally, the reward function gives -1 at each turn to discourage over-confirmation from the system side and promote effectiveness.

3.4.1 Training and Testing Procedures

We use the same DQN architectures with three ways of extracting uncertainty estimates – Bayes-by-backrop, dropout and concrete dropout. Figure 3.1 presents a high-level architecture of a policy model. All deep RL models contain two reLU hidden layers of size 300 and

System Action	Natural Description
request + `slot`	The system requests specific slot from the user
confirm + `slot`	The system asks to confirm the slot from the user
select + `slots`	The system asks the user to choose between some entities
inform + `slots`	The system informs about the entity
repeat	The system repeats the previous system action
reqmore	After providing the entity, the systems asks if it help with anything else
restart	The system restarts the conversation
bye	The system finishes the conversation

Table 3.2 The list of available system actions. The slots from Table 3.1 are combined with dialogue act to a form final representation for some actions.

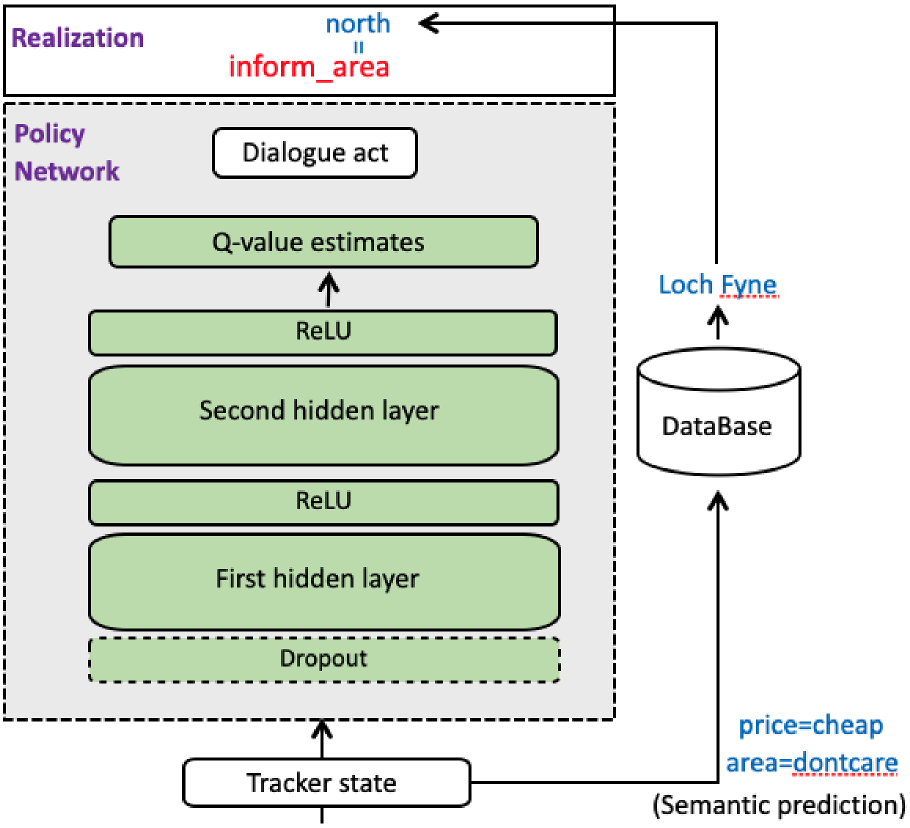


Fig. 3.1 A high-level overview of the deep RL policy network architecture. The input from the tracker state is processed to estimate the Q-value function. Once the action is chosen, the realized dialogue act is combined with information taken from the knowledge base.

100. All models are trained over 4000 simulated dialogues with mini-batches of 64. The experience replay pool size is 6000. Once the first 192 samples are collected, the model is updated every 2 dialogues. Each sample is a state transition (b_t, a_t, r_t, b_{t+1}). After each

200 dialogues the policy parameters are "frozen" and the model is tested for another 200 dialogues. The Adam optimiser is used with a learning rate of 0.001 (Kingma and Ba, 2014). For vanilla DQN, an ε-greedy policy is used, which is initially set to a 0.75 value, and annealed to 0.0 after 4000 training dialogues. All other models follow the Thompson sampling procedure. The dropout starting value is 0.1 for MC-Dropout and 0.2 for Concrete-Dropout. All parameters were chosen based on the grid search results; there have not been substantial differences between different neural architectures except for a one parameter. It is worth noting that we found crucial to carefully set the σ_ε parameter used to sample the covariance parameter σ in the BDQN algorithm.

All deep-RL models are benchmarked against a state-of-the-art non-parametric value-based RL algorithm for dialogue modelling Gaussian Process-SARSA (GPSARSA, Section 2.6.4). GP-SARSA has proven to be highly sample efficient since it can learn from a small number of observations by exploiting the correlations defined by a kernel function. At the same time, it provides an uncertainty measure of its estimates. The Q-value function is modelled as a GP with zero mean 0 and kernel k: $Q(b,a) \sim \mathrm{GP}(0,(k((b,a),(b,a))))$. The Q-value approximation is updated given the collected belief-action pairs (b,a) and their corresponding rewards r (Gašić and Young, 2014). The implicit knowledge of the distance between data points in observation space provided by the kernel greatly speeds up learning since it enables Q-values in as yet unexplored space to be estimated.

3.4.2 Comparison with Baselines

We start the comparison by benchmarking deep RL methods in a simple environment without any errors. This will show whether these methods can obtain similar sample efficiency as GPSARSA at all. In Figure 3.2, we show the average success rate for BDQN, DQN, DQN with dropout, DQN with a concrete dropout and GP-SARSA, in a noise-free environment for both domains. In this set-up, the dialogue manager works with perfect information.

GP-SARSA learns the fastest and is the most stable, benefiting from the ability of Gaussian Processes to learn from a small amount of data, exploiting the correlations defined by the kernel function. The results show that BDQN reaches a performance comparable to GP-SARSA in both domains. Three other analysed methods, dropout, concrete dropout approaches, did not help to improve the sample efficiency over the vanilla ε-greedy algorithm neither they stabilize exploration. Although with concrete dropout tuning of the dropout probability is automatic, it did not help improve efficiency. This is exemplified in a more difficult SFRestaurants domain Figure (3.2, right).

All different settings for α-divergence loss do not perform better than vanilla variational inference loss in general. Convergence to an optimal policy is slower with an increasing

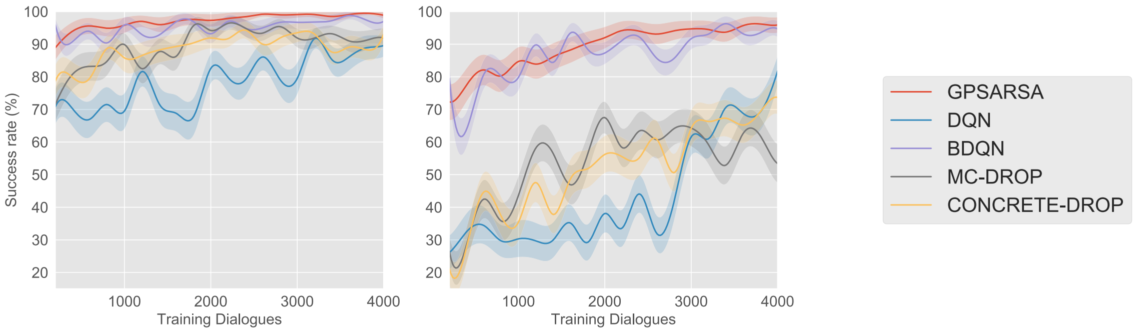


Fig. 3.2 The success rate learning curves for BDQN, GPSARSA, DQN, DQN with dropout and DQN with concrete dropout under noise-free conditions for CamRestaurants and SFRestaurants domains, with two standard error bars.

number of samples. Taking more MC samples decreases the variance of the gradient estimates, and the averaged loss for most updates is closer to the loss obtained when taking a sample close to the mean of the variational distribution q. This implies more updates are necessary to move in the direction of the true posterior distribution p, (by minimizing the $\mathcal{KL}$ or α divergence), trading off for reduced exploration, and slower learning. With the BB-α reparametrization, more than 1 samples are necessary for α-divergence to have any effect, but this reduces exploration efficiency and results in worse performance.

3.4.3 Adaptation to Different Environments

Deployed dialogue systems often experience noisy environments with ASR and NLU error propagating and reaching often 20-50% (Barker et al., 2018). This is modelled by applying a random noise to the semantic input at a pre-defined level; all models are now trained with the simulated user with three different semantic error rates (0, 15, 30). In the case of both domains (Figures 3.3 and 3.4) the higher the confusion rate, the lower the performance. GP-SARSA and BDQN consistently reach the highest performance. The Concrete-Dropout model tends to outperform the MC-Dropout for the more challenging domain. Both of these methods obtain high success rates but they tend to have a lower sample efficiency than BDQN in the initial part of training.

Interestingly, BDQN starts providing the best training efficiency and performance in a more challenging environment with higher error rates. The vanilla DQN algorithm lags substantially behind the models using informative exploration based on uncertainty estimates. The difference in efficiency gets larger for the larger domain.

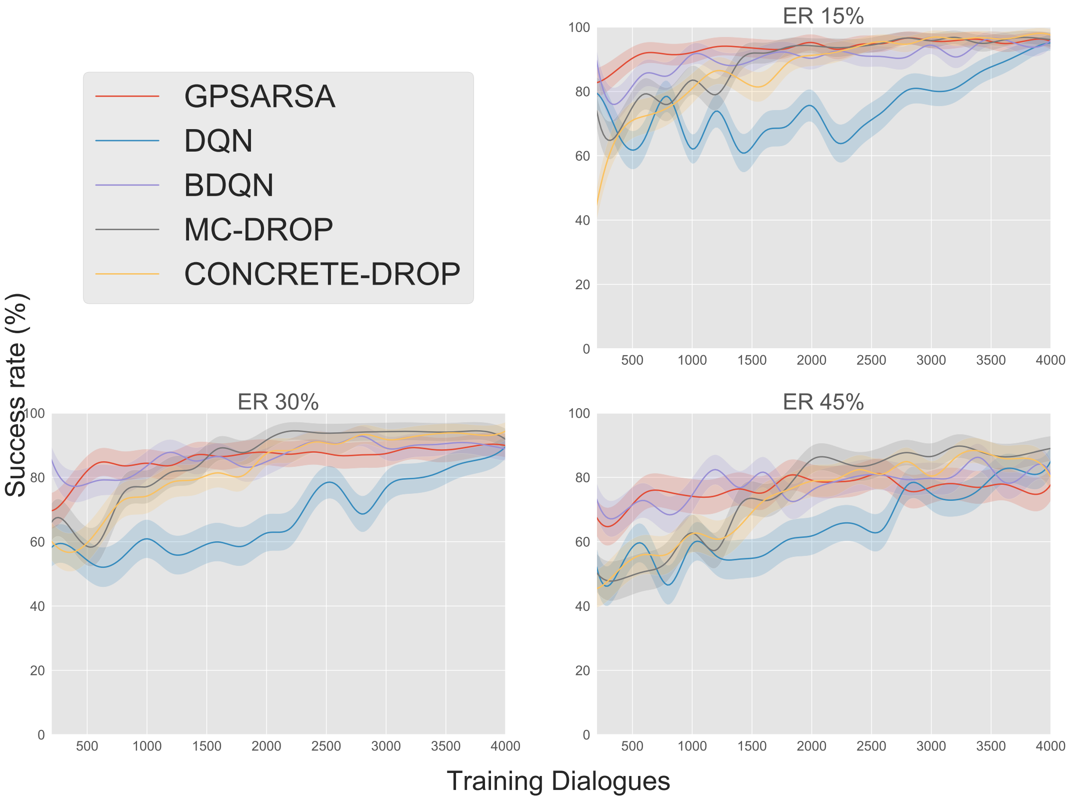


Fig. 3.3 The success rate learning curves for BDQN, GPSARSA, DQN, DQN with dropout and DQN with concrete dropout with different confusion rates in the CamRestaurants domain.

3.4.4 Comparison of Computation Complexity

A straightforward implementation of the GP-based approximation of the Q-value function takes $O(n^3)$ where n is the number of data points due to the computing the inverse of a Gram matrix (Rasmussen and Williams, 2006). Sparse approximations have been proposed to reduce the complexity of exact predictions such as GPSARSA (Engel, 2005). To obtain uncertainty estimates, GPSARSA needs $O(nk^2)$ steps, where n is the total number of data points during training and k is the number of representative data points ($k << n$). The representative points are added along the training only if they significantly contribute to the error of predictions. On the other hand, neural networks scale linearly with the data. Training complexity for standard DQN, dropout and concrete dropout is $O(N)$ in every step where N

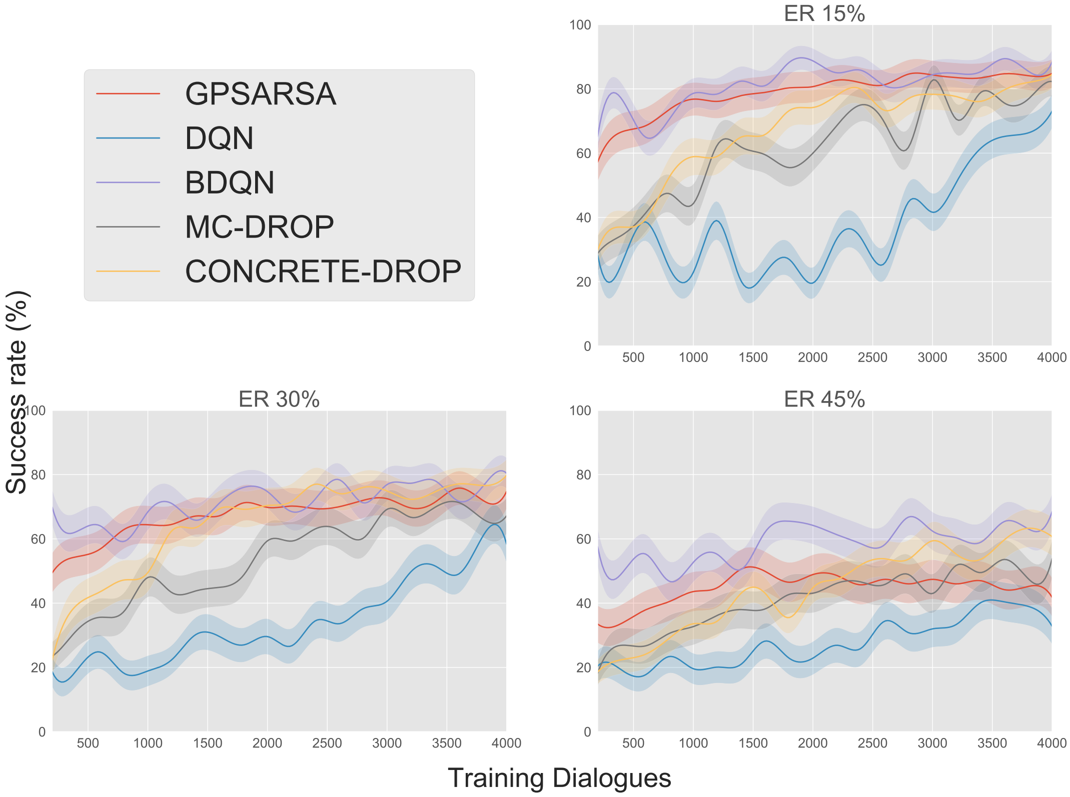


Fig. 3.4 The success rate learning curves for BDQN, GPSARSA, DQN, DQN with dropout and DQN with concrete dropout with different confusion rates in the SFRestaurants domain.

is the number of neural network parameters. Complexity of BDQN is tripled as it requires three sets of parameters. Figures 3.5 shows the training time of two rival methods.

As we can see, with a more challenging domain, the difference in the training time becomes larger due to the increase in the size of the belief state. This difference also happens at the inference time. BDQN achieves a similar (or better sample efficiency) and scales linearly with larger domains unlike the GPSARSA algorithm.

3.4.5 Noise-Robustness In Adversarial Environment

Finally, in order to evaluate the quality and robustness of policy with the simulated user, the agent was trained with a 15% semantic error rate, and then evaluated on a 45% semantic error rate. This examines the generalisation capabilities of different algorithms (Figures 3.6 and

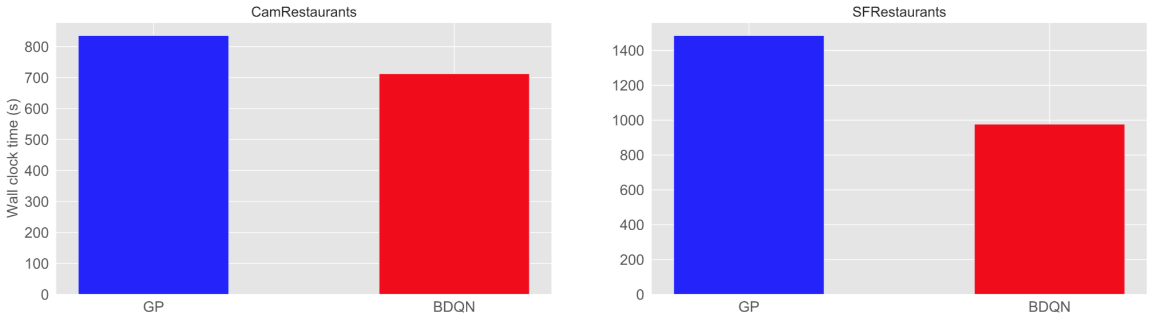


Fig. 3.5 The training time of GP and BDQN methods in the CamRestaurants domain (left) and in the SFRestaurants domain (right). The time is presented in seconds and averaged over 10 runs.

3.7). In this case, BDQN generalizes better than GPSARSA and ε-greedy algorithms. BDQN has the potential for robust performance, and performs well, even at conditions different from the training ones. This is especially observable in the SFRestaurants domain in the beginning of the training (Figure 3.7). Again, the Concrete-Dropout and MC-Dropout models tend to catch-up with the performance of the BDQN method in the final parts of the training. Overall, the result suggests that value-based deep RL optimisation algorithms perform on par against the non-parametric model. This also applies to data efficiency when employed with uncertainty estimates. At the same time, they scale linearly with data which allows for deployment in larger domains with more interactions.

3.5 Conclusion

This chapter has described how Bayesian modelling can be successfully applied in DQN to obtain uncertainty estimates for better dialogue management. The results confirm that BDQN learns dialogue policies with more efficient exploration than ε-greedy based methods, and reach performance comparable to the state-of-the-art in policy optimisation GPSARSA. BDQN is also as sample efficient as GP-SARSA without the computational complexity of GPs. When trained with a noise level of 15% and then evaluated at 45%, BDQN achieved often higher performance at higher confusion rates than standard deep RL method. This shows that BDQN generalizes better than ε-greedy based methods (especially in low data regime) and it is as robust as GP-SARSA. Although MC-Dropout and concrete dropout tend to be less sample efficient than BDQN, they achieved a more stable performance at later stages of training compared to BDQN, and it is a promising avenue to explore.

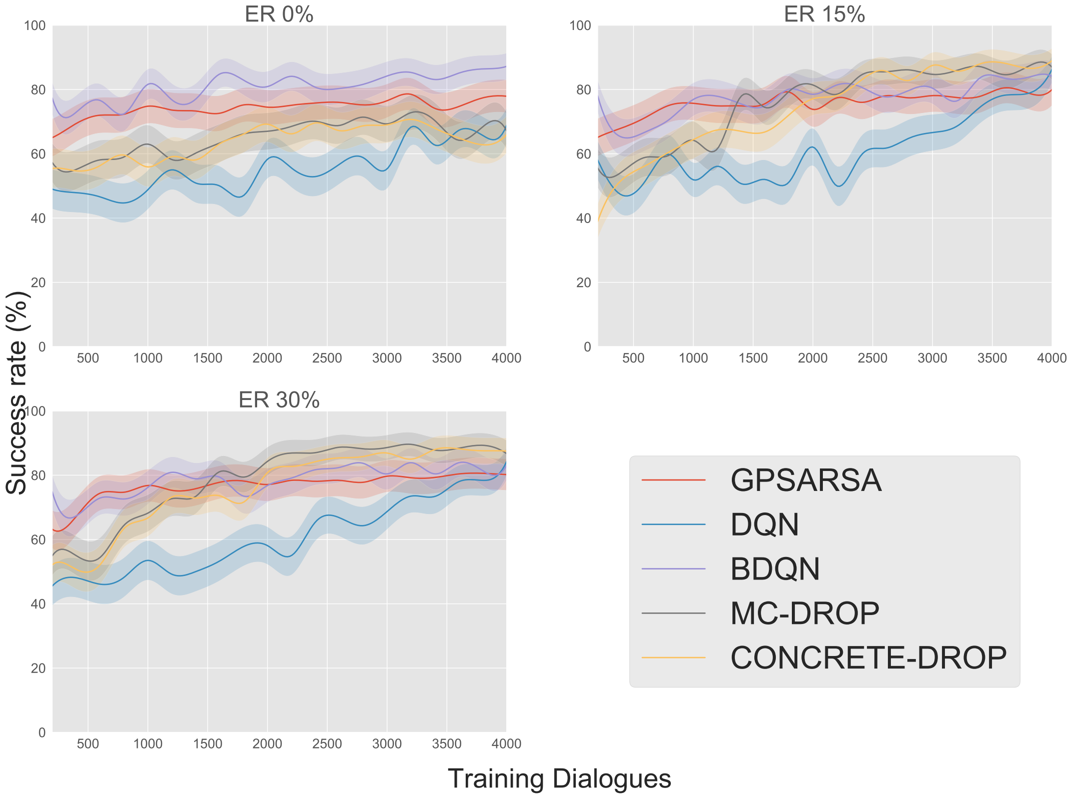


Fig. 3.6 The success rate learning curves for BDQN, GPSARSA, DQN, DQN with dropout and DQN with concrete dropout in CamRestaurants with different confusion rates during training and a 45% confusion rate at testing.

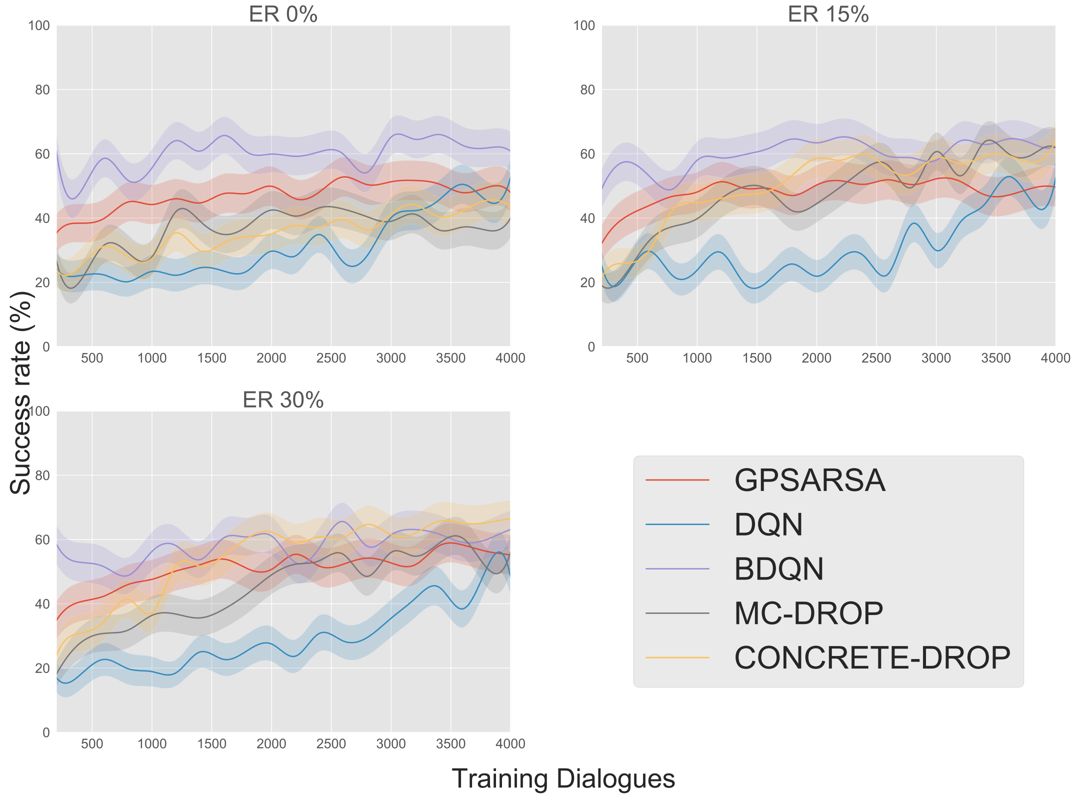


Fig. 3.7 The success rate learning curves for BDQN, GPSARSA, DQN, DQN with dropout and DQN with concrete dropout in SFRestaurants with different confusion rates during training and a 45% confusion rate at testing.

Chapter 4

MultiWOZ - Large Scale Dataset for Multi-Domain Modelling

Prior work in task-oriented modelling was carried out in a low data regime. What is more, policy management does not typically take a direct advantage of learning from any available resources. This chapter addresses these problems by presenting the MultiWOZ corpus. MultiWOZ is a task-oriented large-scale multi-domain dialogue corpus of natural human-human conversations. The chapter starts with the motivation behind the collection of this dataset. Next, a novel and comprehensive overview of common data collection paradigms is presented providing pros and cons of each. We finish this part with a proposal for a new approach. As it entirely relies on crowd-sourcing, it allows for deployment in both research and industrial environments reducing the costs of manual annotations.

The second part of the chapter describes the MultiWOZ corpus. The dataset was collected as a validation process of the proposed data collection procedure. The quantitative and qualitative analysis of the collected corpus is presented. We provide also comparison to currently available resources in the task-oriented domain. This work was first published in (Budzianowski et al., 2018b).

4.1 Motivation

Dialogues systems are inherently hard to build because there are several layers of complexity: the noise and uncertainty in speech recognition (Black et al., 2011); the ambiguity when understanding human language (Williams et al., 2013); the need to integrate third-party services and dialogue context in the decision-making (Paek and Pieraccini, 2008; Traum and Larsson, 2003); and finally, the ability to generate natural and engaging responses

(Stent et al., 2005). These difficulties have led to the same solution of using statistical framework and machine learning for various system components, such as natural language understanding (Henderson et al., 2013), dialogue management (Gašić and Young, 2014), language generation (Wen et al., 2015), and even end-to-end dialogue modelling (Zhao and Eskenazi, 2016).

To drive the progress of building dialogue systems using data-driven approaches, a number of conversational corpora have been released in the past. Based on whether a structured annotation scheme is used to label the semantics, these corpora can be roughly divided into two categories: corpora with structured semantic labels (Asri et al., 2017; Eric et al., 2017; Hemphill et al., 1990; Shah et al., 2018; Wen et al., 2017b; Williams et al., 2013); and corpora without semantic labels but with an implicit user goal in mind (Lowe et al., 2015; Ritter et al., 2010). The former type typically comes with a pre-defined ontology that allows to break down modelling into separate modules (Section 2.1). The latter type lacks well-defined domains, ground-truth knowledge source or user goals. In turn it is not possible to annotate these datasets with required low-level annotation signal. That is why, the significant part of research was historically carried out with the former type of datasets (Black et al., 2011; Lemon and Pietquin, 2007; Walker, 2000; Wen et al., 2017b; Young et al., 2013b, i.a.).

Despite these efforts, aforementioned datasets are usually constrained in one or more dimensions such as missing proper annotations, only available in a limited capacity, lacking multi-domain use cases, or having a negligible linguistic variability. Although data is the driving force behind proposed machine learning models, even in the simplest metric capture i.e. number of dialogues, corpora for task-oriented dialogues are surprisingly small (Table 4.1). This size of data prevents from applying structure-rich and parameter-rich models. For example, the end-to-end models proposed by Wen et al. (2017b) or Lei et al. (2018)

Dataset	**Dialogues**
DSTC2 (Williams et al., 2013)	1,612
SFX (Gašić et al., 2014)	1,006
WOZ2.0 (Wen et al., 2017b)	600
FRAMES (Asri et al., 2017)	1,369
KVRET (Eric et al., 2017)	2,425
M2M (Shah et al., 2018)	1,500

Table 4.1 The comparison of number of dialogues in publicly available corpora for modelling task-oriented dialogues before the collection of our corpus.

contains on average 100k parameters while being trained on around 350 dialogues. It also constraints the application of large pre-trained networks due to over-fitting issues (Cer et al., 2018; Devlin et al., 2018; Radford et al., 2018).

The main focus of this work: learning dialogue policy is particularly affected by this state, i.e. the lack of large conversational corpora. As most of the collected datasets do not contain human-human conversations, the collected dialogues have little resemblance to natural counterparts (Gašić et al., 2014; Williams et al., 2013). Thus, the pre-training phase on available resources skews the model from *optimal* dialogue flows. This breaks the assumption made in Chapter 2. It can be argued that the notion of optimality outside the human-human conversations makes little sense as dialogue is a socially-grounded phenomenon. That is why, dialogue systems should be guided as much as possible by the data signal. Moreover, a robust dialogue policy relies on sufficient coverage of potential dialogue paths. The small size of available corpora limits the access to different dialogue flows especially in the case of complex multi-domain dialogues. The attribution of distinctively human-like behavioural characteristics through imitating human-human dialogues by conversational systems raises an ethical question. However, studies and industry practice show that the human-like behaviour of dialogue systems is strongly preferred by customers (Cohn et al., 2019).

Finally, it can be argued that the introduction of dialogue corpora accelerates and spurs progress in the dialogue modeling. The introduction of the DSTC2 dataset established a Dialogue System Technology Challenges and it facilities the dialogue state tracking research (Henderson et al., 2013; Mrkšić et al., 2017a; Perez and Liu, 2017; Ramadan et al., 2018, i.a). As of writing this work, eight editions of Dialogue System Technology Challenge have been completed (Section 4.6) covering different modelling challenges. Further, the introduction of Cam676 (Wen et al., 2017b) introduced a new wave of research in an end-to-end dialogue modelling (Eric et al., 2017; Lei et al., 2018; Madotto et al., 2018; Wen et al., 2017a, i.a). These examples resembles similar patterns from computer vision (Deng et al., 2009) or automatic speech recognition (Garofolo, 1993).

This chapter tackles three above-mentioned challenges: namely the lack of a 1) large scale task-oriented 2) human-human dialogues corpus covering 3) multiple domains. We introduce a new data-collection paradigm along with the Multi-Domain Wizard-of-Oz (MultiWOZ) dataset. MultiWOZ is a large-scale multi-turn conversational corpus with dialogues spanning across several domains and topics. Each dialogue is annotated with a sequence of dialogue states and corresponding system dialogue acts (Traum, 1999). Hence, MultiWOZ can be used to develop individual system modules as separate classification tasks and serve as a benchmark for existing modular-based approaches. On the other hand, MultiWOZ has around 10k dialogues, which is at least one order of magnitude larger than any structured corpus currently available. This significant size of the corpus allows researchers to carry on end-to-end based dialogue modelling experiments, which may facilitate a lot of exciting ongoing research in the area.

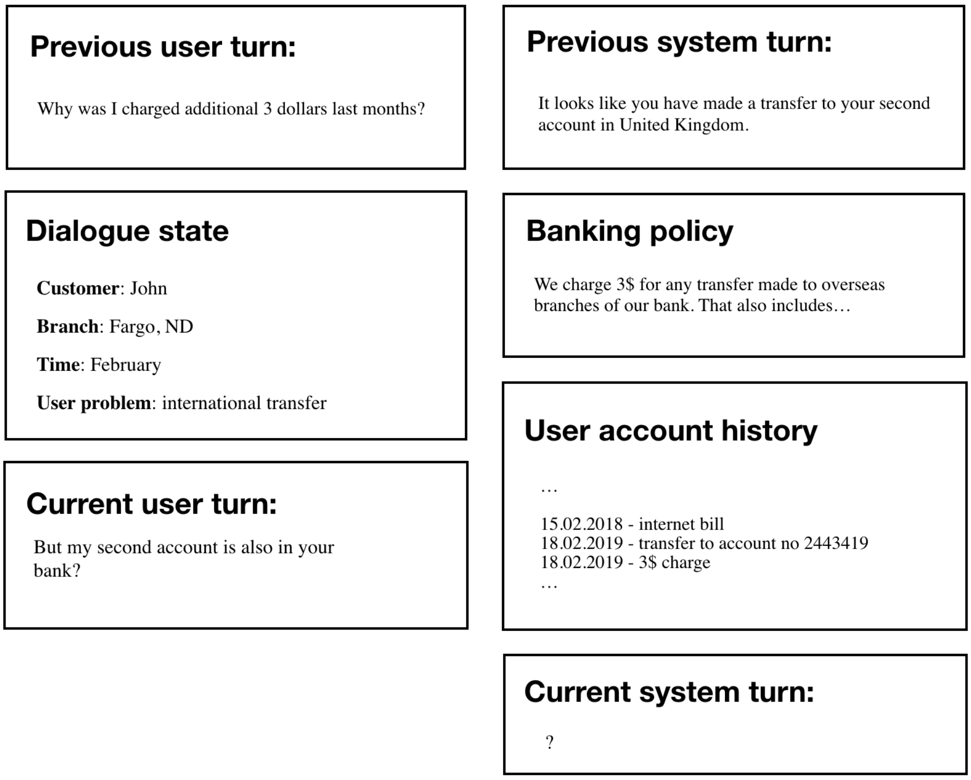
4.2 Overview of Data Collections Paradigms

Unlike object recognition in computer vision or automatic speech recognition, statistical dialogue modeling is a very ambiguous and not well-defined problem simply because it lacks consistent annotations and universal well-defined metrics. Annotations in dialogue datasets are therefore also extremely complex. The dialogue agent faces a variety of challenges while interacting with real users. For instance, the coverage of the conversation is practically infinite as humans can refer to out-of-domain concepts, use metaphors, or rely on the interlocutors' commonsense or general knowledge. The conversation requires both sides to be active across many turns, and it relies on context: concepts introduced at the beginning of a conversation can be referred to much later. Finally, conversations are also grounded in the real world where visual or audio clues serve as an integral conversational context.

Figure 4.1 illustrates the annotation requirements in a hypothetical banking domain over two turns. In order to solve the user's problem, several information signals are required on top of the natural realization of a conversation. Firstly, the annotation over the dialogue state carried across turns are required (DST). The next action of the system is conditioned on the policy of the company (DM). Finally, in order to verify the true state of the user's account, the access to the user's personal history through the database call is needed (DM). All these pieces of information condition the final system response (NLG). The modular modelling of SDS allowed requires different annotation signals for each module and it lead to different data collection paradigms. Existing datasets can be roughly grouped into three categories: machine-to-machine, human-to-machine, and human-to-human conversations. A detailed review of these categories is presented below.

4.2.1 Machine-to-Machine

The Machine-to-Machine paradigm focuses on creating an environment with a simulated user that enables to exhaustively generate dialogue templates. These templates can be mapped to a natural language by either pre-defined rules (the Dialog bAbI corpus) (Bordes et al., 2017) or crowd workers (the M2M corpus)(Shah et al., 2018). The collection of various language expressions helps build better individual components of the conversational agent related to natural language understanding and generation. As the dialogues are created on



Why was I charged additional 3 dollars last months?

It looks like you have made a transfer to your second account in United Kingdom.

Customer: John

Branch: Fargo, ND

Time: February

User problem: international transfer

We charge 3$ for any transfer made to overseas branches of our bank. That also includes…

…
15.02.2018 - internet bill
18.02.2019 - transfer to account no 2443419
18.02.2019 - 3$ charge
…

But my second account is also in your bank?

Fig. 4.1 The hypothetical two turns of a conversation from the customer service domain along with the annotations over the dialogue state, the knowledge base and the user's private information from the database.

the dialogue act level (Section 2.5), we avoid the challenge of labeling ambiguous semantics by simply asking the crowdworkers to generate sentences from the simulated labels. In theory, this paradigm can create an infinite number of conversations to cover all possible in-domain dialogue flows. However, the naturalness of the dialogue flows relies entirely on the engineered set-up of the user and system bots. This poses a risk of a mismatch between training data and real interactions harming the interaction quality. Moreover, these datasets do not take into account noisy conditions often experienced in real interactions (Black et al., 2011). There are also two publicly available software packages based on the M2M paradigm: PyDial (Ultes et al., 2017) and ParlAI (Miller et al., 2017).

4.2.2 Human-to-Machine

Since collecting dialogue corpus for a task-specific application from scratch is difficult, most of the task-oriented dialogue corpora are fostered based on an existing dialogue system. One famous example of this kind is the Let's Go Bus Information System which offers live

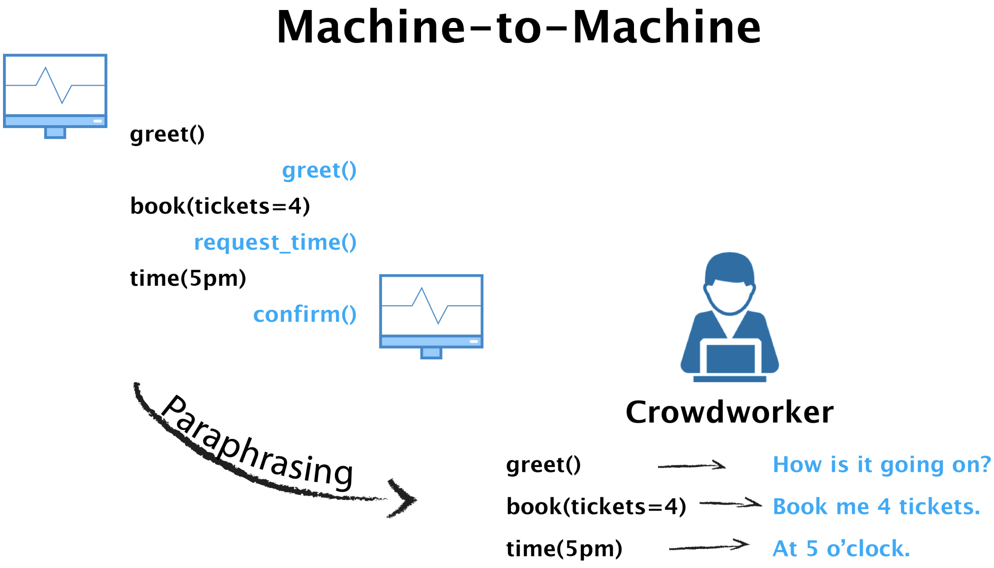


Fig. 4.2 The machine-to-machine paradigms generate first template dialogues with a pre-built user simulator and the dialogue system. The sketches of dialogues are further paraphrased with rules or crowd workers.

bus schedule information over the phone (Raux et al., 2005) leading to the first Dialogue State Tracking Challenge (Williams et al., 2013). Taking the idea of the Let's Go system forward, the second and third DSTCs (Henderson et al., 2014b,c) have produced bootstrapped human-machine datasets for a restaurant search domain in the Cambridge area, UK. Since then, DSTCs have become one of the central research topics in the dialogue community (Kim et al., 2016, 2017). While human-to-machine data collection is an obvious solution for dialogue system development, it is only possible with a provision of an existing working system. Therefore, this chicken (system)-and-egg (data) problem limits the use of this type of data collection to existing system improvement instead of developing systems in a completely new domain. What is even worse is that the capability of the initial system introduces additional biases to the collected data, which may result in a mismatch between the training and testing sets (Wen et al., 2016). The limited understanding capability of the initial system may prompt the users to adapt to simpler input examples that the system can understand but are not necessarily natural in conversations.

4.2.3 Human-to-Human

Arguably, the best strategy to build a natural conversational system may be to have a system that can directly mimic human behaviors through learning from a large amount of real human-human conversations. With this idea in mind, several large-scale dialogue corpora

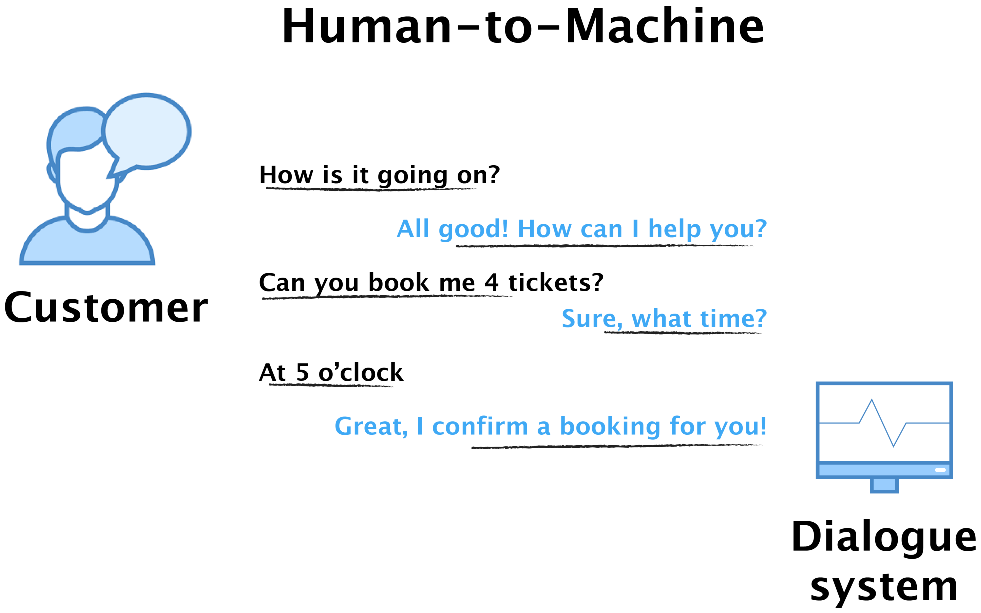

Human-to-Machine
Customer
How is it going on?
All good! How can I help you?
Can you book me 4 tickets?
Sure, what time?
At 5 o'clock
Great, I confirm a booking for you!
Dialogue system

Fig. 4.3 The Human-to-machine (H2M) paradigm requires a pre-built dialogue system. The annotation of system actions comes for free but the dialogues are biased by the system's capacity and the user's susceptibility to interactions.

have been released in the past, such as the Twitter (Ritter et al., 2010) dataset, the Reddit conversations (Schrading et al., 2015), and the Ubuntu technical support corpus (Lowe et al., 2015). Although previous work (Vinyals and Le, 2015) has shown that a large learning system can learn to generate interesting responses from these corpora, the lack of grounding conversations onto an existing knowledge base or APIs limits the usability of developed systems. Due to the lack of an explicit goal in the conversation, recent studies have shown that systems trained with this type of corpus not only struggle in generating consistent and diverse responses (Li et al., 2016a) but are also extremely hard to evaluate (Liu et al., 2016).

However, the human-human conversations for task-oriented domains can be collected in a more constrained environment. The Wizard-of-Oz framework (WOZ) (Kelley, 1984) was first proposed as an iterative approach to improve user experiences when designing a conversational system. The goal of WOZ data collection is to log down the conversation for future system development. One of the earliest dataset collected in this fashion is the ATIS corpus (Hemphill et al., 1990), where conversations between a client and an airline help-desk operator were recorded. Rieser et al. (2014) used the WOZ framework to collect restaurant-search oriented spoken conversations with trained wizards. Recently, Wen et al. (2017b) have shown that the WOZ approach can be applied to collect high-quality typed conversations where a machine learning-based system can learn from. By modifying the original WOZ framework to make it suitable for crowd-sourcing, a total of 676 dialogues

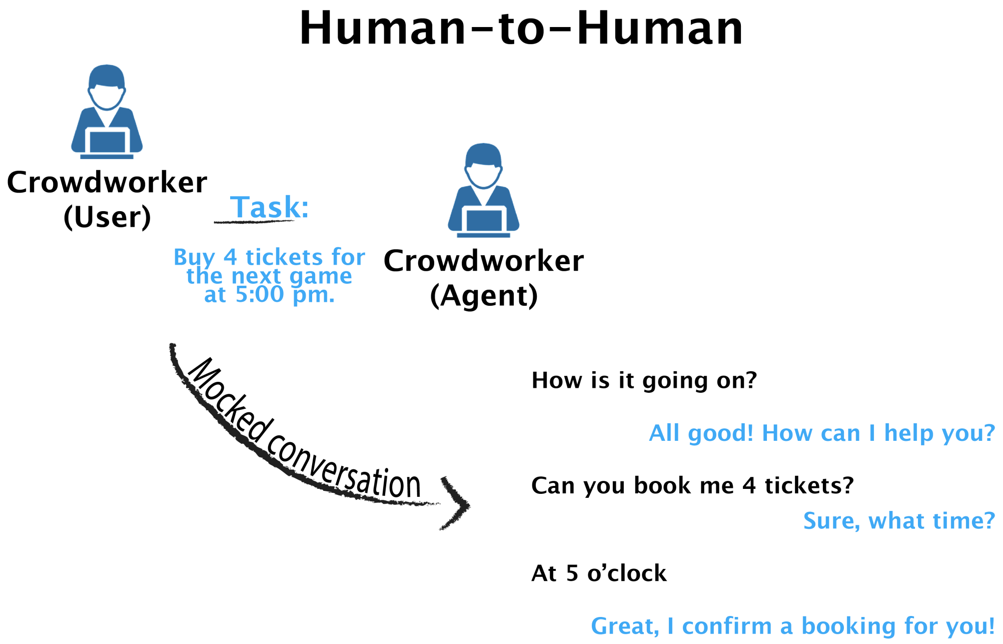

Human-to-Human
Crowdworker (User)
Task:
Buy 4 tickets for the next game at 5:00 pm.
Crowdworker (Agent)
Mocked conversation
How is it going on?
All good! How can I help you?
Can you book me 4 tickets?
Sure, what time?
At 5 o'clock
Great, I confirm a booking for you!

Fig. 4.4 The WOZ data collection setup. The user is given a predefined task while the "system" is equipped with knowledge databases to help fulfill the user's goal. All actions on both sides can be tracked to collect a variety of learning signals stored as rich dialogue annotations.

was collected via Amazon Mechanical Turk. The corpus was later extended to additional two languages for cross-lingual research (Mrkšić et al., 2017b). Subsequently, this approach is followed by Asri et al. (2017) to collect the Frame corpus in a more complex travel booking domain, and Eric et al. (2017) to collect a corpus of conversations for in-car navigation. Despite the fact that all these datasets contain highly natural conversations comparing to other human-machine collected datasets, they are usually small in size with only a limited domain coverage.

4.3 Data Collection Set-up

Following the Wizard-of-Oz set-up (Kelley, 1984), corpora of annotated dialogues can be gathered at relatively low costs and with a small-time effort. This is in contrast to previous approaches (Henderson et al., 2014a) and such WOZ set-up has been successfully validated by Wen et al. (2017b) and Asri et al. (2017). Therefore, we follow the same process to create a large-scale corpus of natural human-human conversations. Our goal was to collect multi-domain dialogues. To overcome the need for relying the data collection to a small set of trusted workers[1], the collection set-up is designed to provide an easy-to-operate system interface for the Wizards and easy-to-follow goals for the users. This results in a bigger diversity and semantical richness of the collected data (see Section 4.4.3). Moreover, having a large set of workers mitigates the problem of artificial encouragement of a variety of behavior from users. A detailed explanation of the data-gathering process from both sides is provided below. Subsequently, we show how the crowd-sourcing scheme can also be employed to annotate the collected dialogues with dialogue acts.

4.3.1 Dialogue Task

The domain of a task-oriented dialogue system is often defined by an ontology, a structured representation of the back-end database. The ontology defines all entity attributes called slots and all possible values for each slot. In general, the slots may be divided into *informable* slots and *requestable* slots. *Informable* slots are attributes that allow the user to constrain the search (e.g., area or price range). *Requestable* slots represent additional information the users can request about a given entity (e.g., phone number). Based on a given ontology spanning several domains, a task template was created for each task through random sampling. This results in single and multi-domain dialogue scenarios and domain-specific constraints were generated. In domains that allowed for that, an additional booking requirement was sampled with some probability.

To model more realistic conversations, goal changes are encouraged. With a certain probability, the initial constraints of a task may be set to values so that no matching database entry exists. Once informed about that situation by the system, the users only needed to follow the goal which provided alternative values.

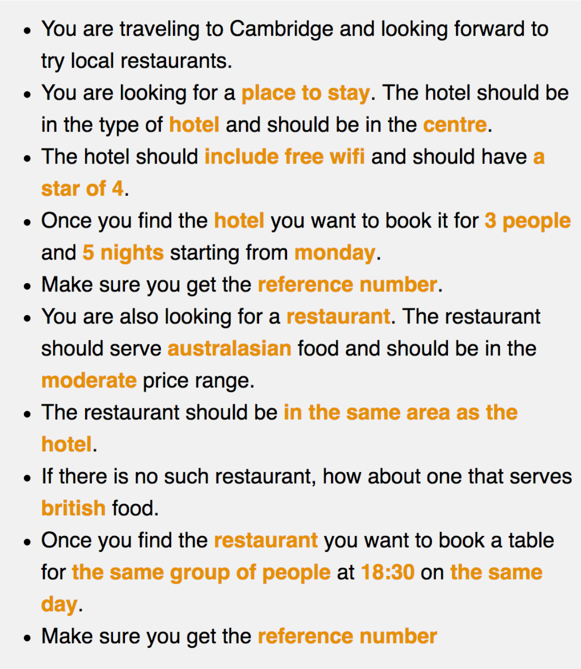
4.3.2 User Side

To provide information to the users, each task template is mapped to natural language. Using heuristic rules, the task is then gradually introduced to the user to prevent an overflow of information. The goal description presented to the user is dependent on the number of turns already performed. Moreover, if the user is required to perform a sub-task (for example – booking a venue), these sub-goals are shown straight-away along with the main goal in the given domain. This makes the dialogues more similar to spoken conversations.[2] Figure 4.5 shows a sampled task description spanning over two domains with booking requirement.

---

[1] Excluding annotation phase.

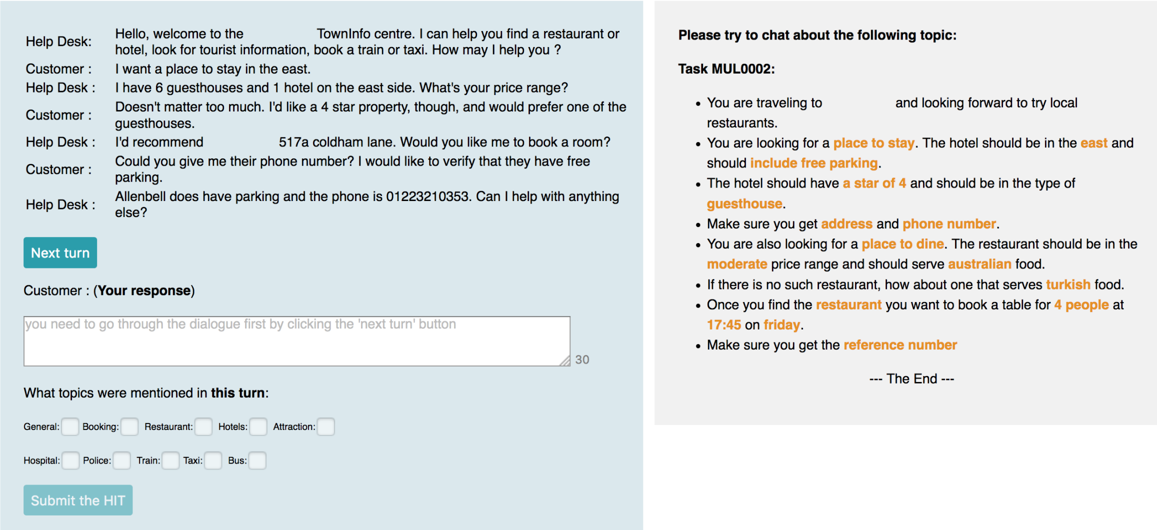
[2] However, the length of turns are significantly longer than with spoken interaction (Section 4.4.3).



- You are traveling to Cambridge and looking forward to try local restaurants.
- You are looking for a **place to stay**. The hotel should be in the type of **hotel** and should be in the **centre**.
- The hotel should **include free wifi** and should have **a star of 4**.
- Once you find the **hotel** you want to book it for **3 people** and **5 nights** starting from **monday**.
- Make sure you get the **reference number**.
- You are also looking for a **restaurant**. The restaurant should serve **australasian** food and should be in the **moderate** price range.
- The restaurant should be **in the same area as the hotel**.
- If there is no such restaurant, how about one that serves **british** food.
- Once you find the **restaurant** you want to book a table for **the same group of people** at **18:30** on **the same day**.
- Make sure you get the **reference number**

Fig. 4.5 A sample task template spanning over three domains: hotels, restaurants and booking. The sub-goals are gradually introduced during the dialogue.

Natural incorporation of co-referencing and lexical entailment into the dialogue was achieved through implicit mentioning of some slots in the goal.



Help Desk: Hello, welcome to the TownInfo centre. I can help you find a restaurant or hotel, look for tourist information, book a train or taxi. How may I help you ?
Customer : I want a place to stay in the east.
Help Desk : I have 6 guesthouses and 1 hotel on the east side. What's your price range?
Customer : Doesn't matter too much. I'd like a 4 star property, though, and would prefer one of the guesthouses.
Help Desk : I'd recommend 517a coldham lane. Would you like me to book a room?
Customer : Could you give me their phone number? I would like to verify that they have free parking.
Help Desk : Allenbell does have parking and the phone is 01223210353. Can I help with anything else?

Customer : (**Your response**)

you need to go through the dialogue first by clicking the 'next turn' button

What topics were mentioned in **this turn**:

General: ☐ Booking: ☐ Restaurant: ☐ Hotels: ☐ Attraction: ☐
Hospital: ☐ Police: ☐ Train: ☐ Taxi: ☐ Bus: ☐

Please try to chat about the following topic:

Task MUL0002:

- You are traveling to and looking forward to try local restaurants.
- You are looking for a **place to stay**. The hotel should be in the **east** and should **include free parking**.
- The hotel should have **a star of 4** and should be in the type of **guesthouse**.
- Make sure you get **address** and **phone number**.
- You are also looking for a **place to dine**. The restaurant should be in the **moderate** price range and should serve **australian** food.
- If there is no such restaurant, how about one that serves **turkish** food.
- Once you find the **restaurant** you want to book a table for **4 people** at **17:45** on **friday**.
- Make sure you get the **reference number**

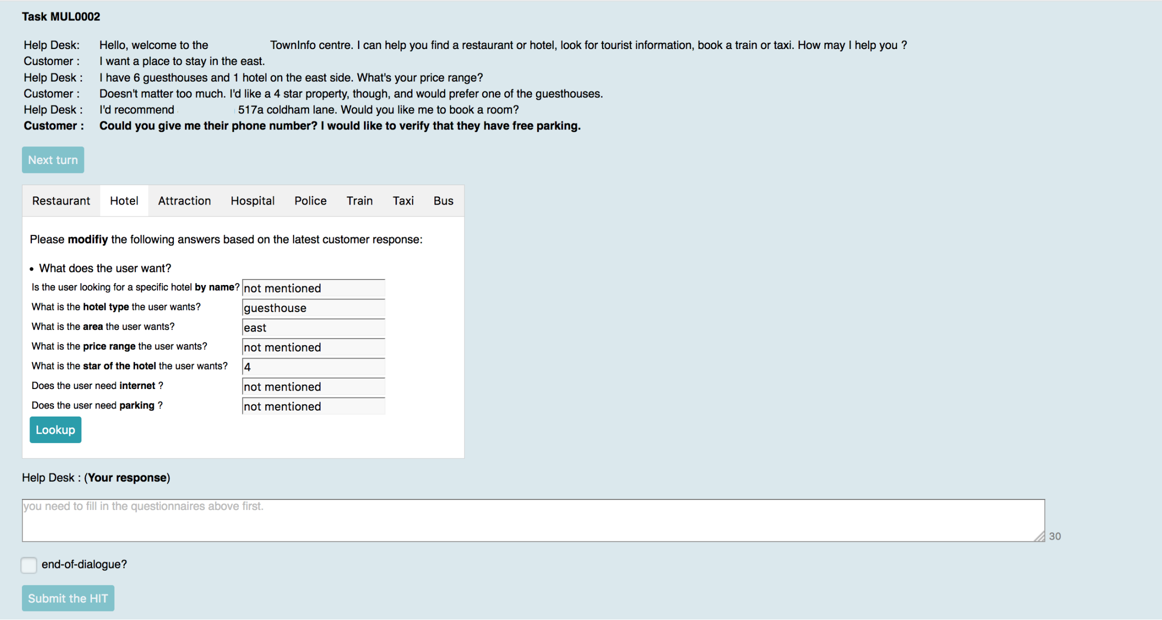
--- The End ---

Fig. 4.6 The interface of the user side. The crowd-worker has to first read the dialogue history. Then, the response box becomes available. The user is also asked to annotate the topics mentioned in the current turn.

Figure 4.6 presents the user side interface where the worker needs to properly respond given the task description and the dialogue history.

4.3.3 System Side

The wizard is asked to perform the role of a clerk by providing information required by the user. He is given an easy-to-operate graphical user interface to the back-end database. The wizard conveys the information provided by the current user input through a web form. This information is persistent across turns and is used to query the database. Thus, the annotation of a belief state is performed implicitly while the wizard is allowed to fully focus on providing the required information. Given the result of the query (a list of entities satisfying current constraints), the wizard either requests more details or provides the user with adequate information. At each system turn, the wizard starts with the results of the query from the previous turn. Figure 4.7 shows the wizard page with the GUI over all domains.



Task MUL0002

Help Desk: Hello, welcome to the TownInfo centre. I can help you find a restaurant or hotel, look for tourist information, book a train or taxi. How may I help you ?
Customer : I want a place to stay in the east.
Help Desk : I have 6 guesthouses and 1 hotel on the east side. What's your price range?
Customer : Doesn't matter too much. I'd like a 4 star property, though, and would prefer one of the guesthouses.
Help Desk : I'd recommend 517a coldham lane. Would you like me to book a room?
Customer : Could you give me their phone number? I would like to verify that they have free parking.

Please **modifiy** the following answers based on the latest customer response:

- What does the user want?

Is the user looking for a specific hotel **by name**?	not mentioned
What is the **hotel type** the user wants?	guesthouse
What is the **area** the user wants?	east
What is the **price range** the user wants?	not mentioned
What is the **star of the hotel** the user wants?	4
Does the user need **internet** ?	not mentioned
Does the user need **parking** ?	not mentioned

Help Desk : (**Your response**)

you need to fill in the questionnaires above first.

☐ end-of-dialogue?

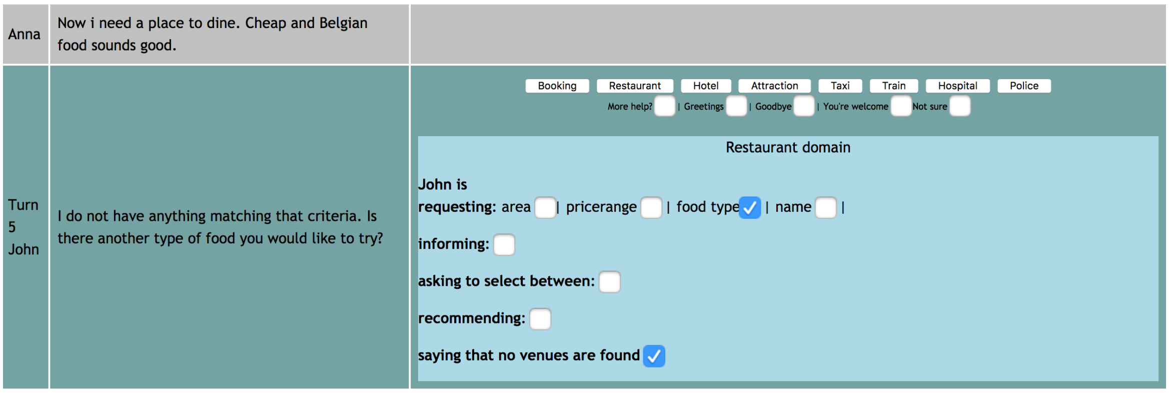
Fig. 4.7 The interface of the wizard side. The crowd-worker has to first read the dialogue history. Then, the window with a list of databases activates. The crowd-worker updates the information given by the customer. The lookup button sends the query to the database and provides the list of entities satisfying user requirements. Finally, the response box becomes available.

To ensure coherence and consistency, the wizard and the user alike first need to go through the dialogue history to establish the respective context. We found that even though multiple workers contributed to one dialogue, only a small margin of dialogues was incoherent.

4.3.4 Annotation of Dialogue Acts

Arguably, the most challenging and time-consuming part of any dialogue data collection is the process of annotating dialogue acts. One of the major challenges of this task is the definition of a set and structure of dialogue acts (Bunt, 2006; Traum and Hinkelman, 1992). In general, a dialogue act consists of the intent (such as request or inform) and slot-value pairs. For example, the act `inform(domain=hotel,price=expensive)` has the intent *inform*, where the user is informing the system to constrain the search to expensive hotels.

Expecting a big discrepancy in annotations between annotators, we initially ran three trial tests over a subset of dialogues using Amazon Mechanical Turk. Three annotations per dialogue were gathered resulting in around 750 turns. As this requires a multi-annotator metric over a multi-label task, we used Fleiss' kappa metric (Fleiss, 1971) per single dialogue act. Although the weighted kappa value averaged over dialogue acts was at a high level of 0.704, we have observed many cases of very poor annotations and an unsatisfactory coverage of dialogue acts. Initial errors in annotations and suggestions from crowd workers gradually helped us to expand and improve the final set of dialogue acts from 8 to 13 – see Table 4.2.


Booking
Restaurant
Hotel
Attraction
Taxi
Train
Hospital
Police
More help? ☐ | Greetings ☐ | Goodbye ☐ | You're welcome ☐ Not sure ☐

Anna: Now i need a place to dine. Cheap and Belgian food sounds good.

Turn 5 John: I do not have anything matching that criteria. Is there another type of food you would like to try?

Restaurant domain

John is
requesting: area ☐ | pricerange ☐ | food type ☑ | name ☐ |
informing: ☐
asking to select between: ☐
recommending: ☐
saying that no venues are found ☑

Fig. 4.8 The annotation set-up for one turn. First, the crowd-worker chooses the domain that the current dialogue concerns. Then, she finishes the suggested sentences annotating dialogue acts with slot and values

The variation in annotations made us change the initial approach. We ran a two-phase trial to first identify a set of workers that perform well. Turkers were asked to annotate an illustrative, long dialogue which covered many problematic examples that we have observed in the initial run described above. All submissions that were of high quality were inspected and corrections were reported to annotators. Workers were asked to re-run a new trial dialogue. Having passed the second test, they were allowed to start annotating real dialogues. This procedure resulted in a restricted set of annotators performing high-quality annotations.

Figure 4.8 shows the set-up for annotation of the system acts with Restaurant domain being turned on.

4.3.5 Data Quality

Data collection was performed in a two-step process. First, all dialogues were collected and then the annotation process was launched. This setup allowed the dialogue act annotators to also report errors (e.g., not following the task or confusing utterances) found in the collected dialogues. As a result, many errors could be corrected. Finally, additional tests were performed to ensure that the provided information in the dialogues matches the pre-defined goals. To estimate the inter-annotator agreement, the averaged weighted kappa value for all dialogue acts was computed over 291 turns. With $\kappa = 0.884$, an improvement in agreement between annotators was achieved although the size of the action set was significantly larger.

4.4 The MultiWOZ Dialogue Corpus

The main goal of the data collection was to acquire highly natural conversations between a tourist and a clerk from an information center in a touristic city. We considered various possible dialogue scenarios ranging from requesting basic information about attractions through booking a hotel room or travelling between cities. In total, the presented corpus consists of 7 domains – *Attraction, Hospital, Police, Hotel, Restaurant, Taxi, Train*. The latter four are extended domains which include the sub-task *Booking*. Through a task sampling procedure (Section 4.3.1), the dialogues cover between 1 and 5 domains per dialogue thus greatly varying in length and complexity. This broad range of domains allows to create scenarios where domains are naturally connected. For example, a tourist needs to find a hotel, to get the list of attractions and to book a taxi to travel between both places. Table 4.2 presents the global ontology with the list of considered dialogue acts.

4.4.1 Data Statistics

Following the data collection process from the previous section, a total of $10,438$ dialogues were collected. Figure 4.9 (left) shows the dialogue length distribution grouped by single and multi-domain dialogues. Around 70% of dialogues have more than 10 turns which shows the complexity of the corpus. The average number of turns is 8.93 and 15.39 for single and multi-domain dialogues respectively with $115,434$ turns in total. Figure 4.9 (right) presents a distribution over the turn lengths. As expected, the wizard replies are much longer – the average sentence lengths are 11.75 and 15.12 for users and wizards respectively.

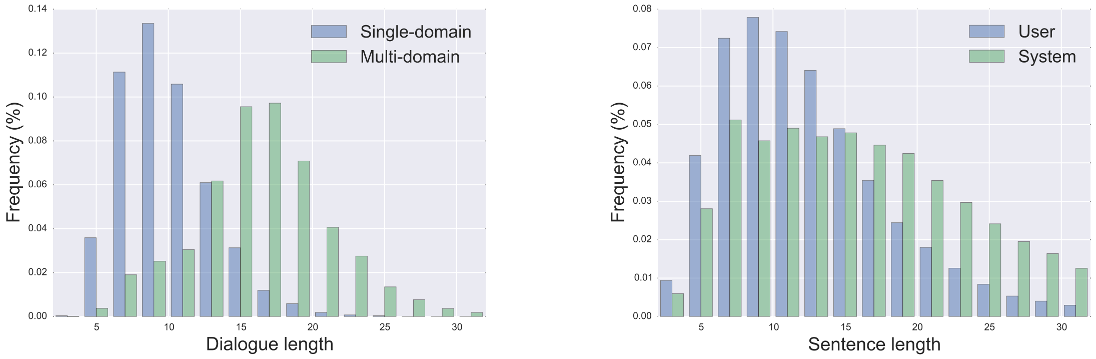


Fig. 4.9 Dialogue length distribution (left) and distribution of number of tokens per turn (right).

The responses are also more diverse thus enabling the training of more complex generation models.

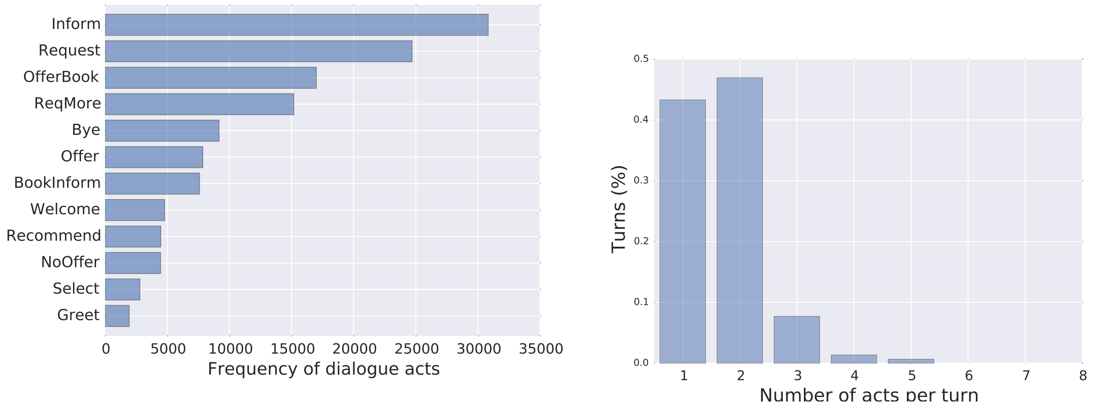


Fig. 4.10 Dialogue acts frequency (left) and number of dialogue acts per turn (right) in the collected corpus.

Figure 4.10 (left) shows the distribution of dialogue acts annotated in the corpus. We present here a summarized list where different types of actions like *inform* are grouped together. The right graph in Figure 4.10 presents the distribution of number of acts per turn. Almost 60% of dialogues turns have more than one dialogue act showing again the richness of system utterances. These create a new challenge for reinforcement learning-based models requiring them to operate on concurrent actions.

In total, $1,249$ workers contributed to the corpus creation with only few instances of intentional wrongdoing. Additional restrictions were added to automatically discover instances of very short utterances, short dialogues or missing single turns during annotations. All such cases were corrected or deleted from the corpus.

4.4.2 Data Structure

There are 3,406 single-domain dialogues that include booking if the domain allows for that and 7,032 multi-domain dialogues consisting of at least 2 up to 5 domains. To enforce reproducibility of results, the corpus was randomly split into a train, test and development set. The test and development sets contain 1k examples each. Even though all dialogues are coherent, some of them were not finished in terms of task description. Therefore, the validation and test sets only contain fully successful dialogues thus enabling a fair comparison of models.

act type	inform / request / select / recommend/ / not found request booking info / offer booking / inform booked / decline booking welcome /greet / bye / reqmore
slots	address / postcode / phone / name / no of choices / area / pricerange / type / internet / parking / stars / open hours / departure destination / leave after / arrive by / no of people / reference no. / trainID / ticket price / travel time / department / day / no of days

Table 4.2 Full ontology for all domains in our data-set. The upper script indicates which domains it belongs to: *: universal, 1: restaurant, 2: hotel, 3: attraction, 4: taxi, 5: train, 6: hospital, 7: police.

Each dialogue consists of a goal, multiple user and system utterances as well as a belief state and set of dialogue acts with slots per turn. Additionally, the task description in natural language presented to turkers working from the visitor's side is added.

4.4.3 Comparison to Other Structured Corpora

To illustrate the contribution of the new corpus, we compare it on several important statistics with the DSTC2 corpus (Henderson et al., 2014a), the SFX corpus (Gašić et al., 2014), the WOZ2.0 corpus (Wen et al., 2017b), the FRAMES corpus (Asri et al., 2017), the KVRET corpus (Eric et al., 2017), and the M2M corpus (Shah et al., 2018). Figure 4.3 clearly shows that our corpus compares favorably to all other data sets on most of the metrics with the number of total dialogues, the average number of tokens per turn and the total number of unique tokens as the most prominent ones. Especially the latter is important as it is directly linked to linguistic richness.

Metric	**DSTC2**	**SFX**	**WOZ2.0**	**FRAMES**	**KVRET**	**M2M**	**MultiWOZ**
# Dialogues	1,612	1,006	600	1,369	2,425	1,500	**8,438**
Total # turns	23,354	12,396	4,472	19,986	12,732	14,796	**113, 556**
Total # tokens	199,431	108,975	50,264	251,867	102,077	121,977	**1,490,615**
Avg. turns per dialogue	14.49	12.32	7.45	**14.60**	5.25	9.86	13.46
Avg. tokens per turn	8.54	8.79	11.24	12.60	8.02	8.24	**13.13**
Total unique tokens	986	1,473	2,142	12,043	2,842	1,008	**23689**
# Slots	8	14	4	**61**	13	14	24
# Values	212	1847	99	3871	1363	138	**4510**

Table 4.3 Comparison of our corpus to other largest task-oriented data sets. Numbers in bold indicate best value for the respective metric. The numbers are provided for the training part of data except for FRAMES data-set were such division was not defined.

4.5 A New Wave of Datasets and End-to-End Modelling

The collection and release of MultiWOZ pushed further the research in two exciting areas of dialogue modelling. First, last year witnessed a substantial increase in the availability of task-oriented resources influenced by MultiWOZ. Secondly, it facilitated the end-to-end dialogue modelling as it was adopted in the latest Dialogue System Challenge.

4.5.1 New Data Collection Paradigms and Task-Oriented Corpora

After the release of MultiWOZ, both academic and industrial research groups have built upon our proposed paradigm leading to the introduction of several new dialogue corpora. Byrne et al. (2019) carried the collection combining the WOZ paradigm with “self-dialogue” in which crowd-sourced workers write the entire dialogue themselves. Contrary to MultiWOZ, crowd-workers were not restricted to any knowledge-base but trained annotators labelled dialogues with API calls and arguments after the initial collection. The Schema-Guided Dialogue (SGD) dataset uses the M2M paradigm to focus on the problem of supporting a larger number of services than in MultiWOZ (Rastogi et al., 2019). Rather than creating one large unified schema for the assistant, each service provides a schema listing the supported slots and intents along with their natural language descriptions. The assistant employs a single unified model containing no domain or service-specific parameters to make predictions conditioned on these schema elements. Finally, the community witnessed a substantial increase in the size of available corpora. The MetalWOZ dataset introduces almost 40000 human-human dialogues covering 47 domains collected with the WOZ paradigm. The primary role of the dataset is to test transfer learning capabilities across new domains (Lee et al., 2019). The MultiDoGO corpus introduces over 80000 dialogues collected through the

Wizard-of-Oz approach wherein a crowd-sourced worker is paired with a trained annotator (Peskov et al., 2019). The dataset differentiates between the agents' speech acts annotated with generic class labels common across all domains, while customer speech acts are labeled with intent classes.

4.5.2 Dialogue System Technology Challenge 8 - Multi-Domain Task Completion Track

MultiWOZ was also adopted as a common benchmark for the end-to-end dialogue modelling. The latest 8th edition of the Dialogue System Technology Challenge introduced the end-to-end task completion track based on MultiWOZ (Kim et al., 2019). The challenge intends to foster progress in two aspects of dialogue systems: dialogue complexity and scaling to a new domain. In order to mitigate the problem of evaluation (Liu et al., 2016), the challenges offer three different evaluation approaches: 1) corpus-based evaluation including slot state accuracy, joint state accuracy, 2) simulation-based evaluation and 3) crowd worker-based evaluation. Along with the challenge, an open-source multi-domain end-to-end dialogue system platform was introduced (Lee et al., 2019). It is the first open-source project that allows to compare and merge conventional pipeline systems with end-to-end neural models in common environments. The track attracted 12 teams that submitted a wide range of models and approaches. The result of the challenge highlights two important aspects of dialogue modelling (Li et al., 2020). Although the best submissions achieved a very high task-completion rate with automatic metrics, the human evaluation shows that the multi-domain dialogues, even in highly constrained domains, still pose a significant challenge. Secondly, the challenge proved that the end-to-end models can perform on par with rule-based systems, a significant result that shows promise for more data-driven dialogue managers.

4.6 Conclusion

As more and more speech oriented applications are commercially deployed, the necessity of building an entirely data-driven conversational agent becomes more apparent. Various corpora were gathered to enable data-driven approaches to dialogue modelling. To date, however, available datasets were usually constrained in linguistic variability or lacking multi-domain use cases. In this chapter, we established a data-collection pipeline entirely based on crowd-sourcing enabling to gather a large scale, linguistically rich corpus of human-human conversations. The crowsourced effort proved that the intuitive set-up of user interface and the gradual introduction of the conversation's goal allows to collect rich and coherent dialogues spanning several domains in concurrent way. MultiWOZ offers valuable training data and a new challenging test-bed for existing modular-based approaches ranging from belief tracking to dialogue acts generation. Moreover, the scale of the data should help to push forward research in the end-to-end dialogue modelling.

Chapter 5

Towards End-to-End Multi-Domain Dialogue Modelling

The mere size of the data, although important from the perspective of training parameter-rich models, is not a fundamental game-changer. In this chapter, using a set of three standard tasks in dialogue modelling: 1) dialogue state tracking, 2) policy optimisation and 3) dialogue-act-to-text generation (NLG) we show that MultiWOZ enables modelling challenges of long-term task-oriented conversations. In the first part of the chapter, we focus on building neural dialogue manager generalizing existing architectures to multi-domain dialogues. The proposed model combines the dialogue policy logic with generation and it illustrates the complexity of the new dataset. Finally, the results for DST and NLG are presented showing further the usability of the data in a variety of different dialogue problems and sets a baseline for future studies.

A substantial part of this chapter is based on work published in (Budzianowski et al., 2018a). The results from sections 5.4 have been a joint work with Bo-Hsieng Tseng and Osman Ramadan as a part of his master's project published in (Ramadan et al., 2018).

5.1 Motivation

Statistical approaches to dialogue modelling typically comprise of various statistical components. This includes a spoken language understanding module, which takes a sentence as input and gives a dialogue act as output, a dialogue belief state tracker that predicts user intent and track the dialogue history, a dialogue policy to determine the dialogue flow, and a natural language generator to convert conceptual representations into system responses. At the other end of the spectrum, sequence-to-sequence learning (Sutskever et al., 2014) has inspired several efforts to build end-to-end trainable conversational systems (Li et al., 2016b; Serban et al., 2015). This family of approaches treats dialogue as a source to target sequence transduction problem, applying an encoder network (Cho et al., 2014) to encode a user query into a distributed vector representing its semantics, which then conditions a decoder network to generate each system response.

The collection of large-scale corpus allows to revisit the idea of bootstrapping the policy module with dialogue corpora (Section 2.6.5). As the conversations involved only humans, the naturalness and efficiency ensure the optimality of dialogue flows. This is in stark contrast to corpora where dialogue system takes an active part in data collection (Gašić et al., 2014; Williams et al., 2013) (Sections 4.2.1 and 4.2.2). Moreover, Lemon (2011) argued for, and empirically demonstrated, the benefit of jointly optimizing dialogue management and natural language generation, within a statistical framework.[1] Recent works (Eric et al., 2017; Lei et al., 2018; Wen et al., 2017b) validates this approach using function approximation with direct generation of natural language. The merge of dialogue management with generation creates several benefits over a standard, modular approach:

1. The policy module can be seamlessly bootstrapped with natural conversations simply bypassing explicit dialogue act annotations if they are missing. This significantly increases the amount of available data.

2. The complexity of system utterances in the multi-domain setting increases significantly (Figures 4.10) and 4.9) (Rieser and Lemon, 2009; Wen et al., 2016). Merging both of the modules allows to model more fine-grained domain nuances relying on latent cues learnt from data rather than explicitly defining a more complex system act taxonomy.

3. The modelling labour is greatly reduced by maintaining one model without the need for establishing a communication channel between DM and NLG.

Moreover, the first applications of neural generative models in the dialogue context produced results far from any real-world applications (Li et al., 2016b; Serban et al., 2017; Vinyals and Le, 2015). However, neural language generation recently made a significant progress thanks to pre-training (Howard and Ruder, 2018; Radford et al., 2019). The increase in quality and coherence of generation shows that modelling directly dialogue reasoning through natural realization is not far off. Also, transformer architecture shows the ability to learn new (i.e., domain-specific) token embeddings in the fine-tuning phase (Radford et al., 2019; Wolf et al., 2019). This means that the powerful generative models can adapt through special

---

[1] The proposed model divides the policy between a high-level dialogue manager and a sentence planner that is learnt as a sub-policy.

tokens to particular tasks. By providing the input representation as text with domain-specific tokens, we could potentially use off-the-shelf architectures and adapt to the domain-specific input without the need of training new dialogue sub-modules. This is of high importance for highly-domain specific dialogues considered in this work. Finally, the long-standing problem of dull and repetitive response generation (Li et al., 2016b) has been in the focus of recent work (Holtzman et al., 2019; Kulikov et al., 2018). Owing to new sampling strategies, generative models are now able to create longer and more coherent sequence outputs. It is worth emphasizing that the merge of DM and NLG modules does not preclude from using low-level annotation signals if available. We will take advantage of it in Chapter 6.

5.2 Multi-domain Neural Dialogue Manager

Modelling multi-domain dialogue management has been severely affected by the lack of complex, long-term task-oriented dialogues (Gašić et al., 2015; Lison, 2011; Peng et al., 2017; Wang et al., 2014). The proposed models re-engineered multi-domain conversations through creating artificial environments often without the natural realization layer (Budzianowski et al., 2017; Cuayáhuitl, 2009; Wang et al., 2014). The collected corpus creates the opportunity to model complex dialogues with validation in the data. Existing approaches maintain separate dialogue managers for either dialogue domains (Gašić et al., 2015) or sub-policies (Budzianowski et al., 2017; Peng et al., 2017). In lieu of recent advancements in building larger and of higher capacity structures (Howard and Ruder, 2018; Radford et al., 2018, 2019), it can be argued that only one dialogue manager should carry out reasoning. This allows to seamlessly switch between the domains and re-use dialogue policies. For example, sub-routines like booking have very similar dialog flows regardless of the domain. In each case slots related with date and time are required along with issuing an API call to reserve the booking. Furthermore, due to the nature of task-oriented dialogues, data scarcity will always be a permanent problem. Therefore, a one structure that does not have to re-learn syntax and grammar-related characteristic of languages increase the sample efficiency and scalability across new domains.

Advances in sequence to sequence learning (Sutskever et al., 2014) has inspired several efforts to build end-to-end trainable, task-oriented conversational systems where modular approach (Young et al., 2013b) can be replaced with one neural architecture. This approach should in theory reduce the cost of building and maintaining task-oriented systems and ease-up transfer learning between different domains. (Wen et al., 2017b) proposed a seminal neural network-based model for task-oriented dialogue systems by balancing the strengths and the weaknesses of the two research communities. The model is end-to-end trainable but modularly connected; it does not directly model the user goal, but nevertheless, it still learns to accomplish the required task by providing relevant and appropriate responses at each turn. The model consists of a separate belief tracker, a database pointer which is based on results from the understanding, and policy and generation modules. Several models built upon this baseline tackling some of its inefficiencies. (Eric et al., 2017) proposed to combine attentions over the input sequence and the database. The output probabilities from both sources are combined to form the generated answer. Madotto et al. (2018) combines the multi-hop attention over memories (Sukhbaatar et al., 2015) with the idea of pointer network (Vinyals et al., 2015). The model incorporates the database into dialogue context and it uses global multi-hop attention mechanisms to copy words directly from dialog history or KBs. These architectures do not break differentiability, however, they typically require a large amount of data to train (Bordes et al., 2017; Eric et al., 2017). Moreover, the performance does not scale well with the increase in the size of the database. All of above-mentioned models were created and evaluated on simple and single-domain oriented datasets. Most of dialogues have around 3 system turns making this more similar to a question-answering set-up. Also, they often involve the pre-training of some of their components. The proposed dialogue manager significantly expands on a number of domains and complexity of conversations that it can handle.

5.3 Evaluating the state of the art model on MultiWOZ

We experimented with evaluating a neural response generation model with an *oracle* belief-state obtained from the wizard annotations (Section 2.5). This allows us to obtain a clear benchmark where the performance of the composite of dialogue management and response generation is completely independent of the belief tracking. In the creation of our multi-domain neural dialogue manager we follow the state of the art model for a single domain Wen et al. (2017b) who frames the dialogue as a context to response mapping problem. The structure of the model is adapted to the multi-domain environment where the system can interchangeably switch between various domains and services. An architecture of the multi-domain dialogue system is composed of three sources of input. 1) the oracle belief state, 2) the database pointer and 3) the dialogue context. These three encoded signals are concatenated and transformed by the feed-forward network to serve as an initial context for the decoder. Figure 5.1 presents a high-level outline of the system. We consider a dialogue that consists of T turns. Where it does not create confusion, L denotes a length of both user and system sentences per turn. The full ontology consists of $\mathcal{D}$ domains, each with $\mathcal{S}_d$ slots and $\mathcal{V}_d$ values. We differentiate between a user output (u) and a machine (system) output (m).

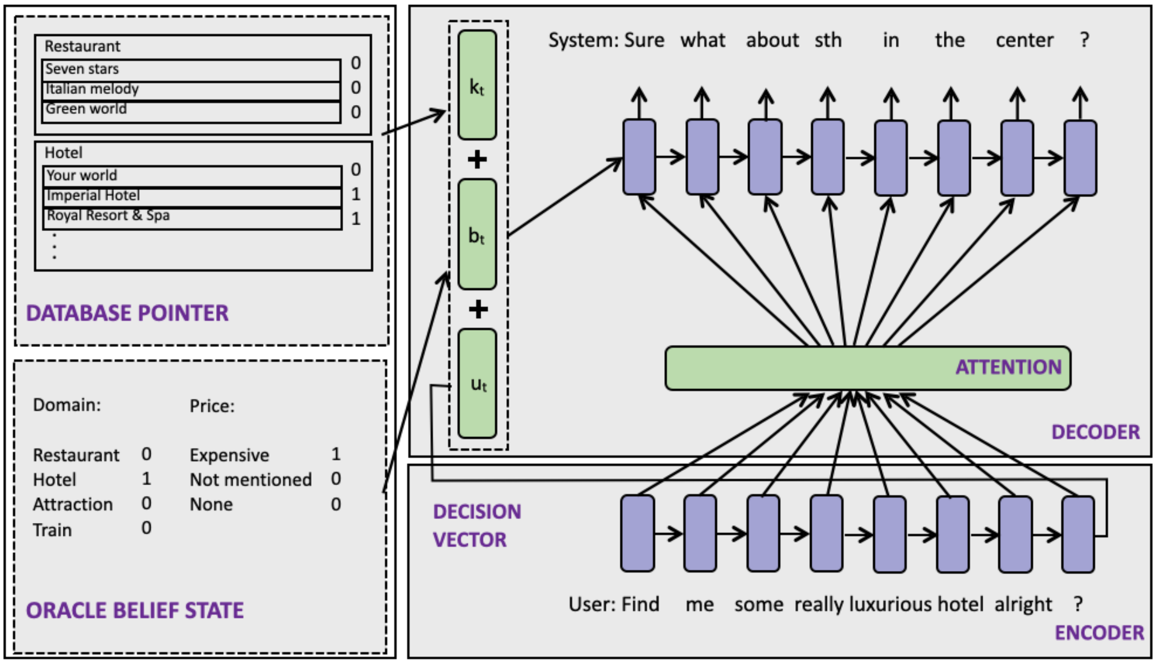


Fig. 5.1 An architecture of the multi-domain dialogue system composed of three sources of input: 1) the oracle belief state, 2) the database pointer and 3) the dialogue context.

5.3.1 Intent Modelling

At each turn, the encoder takes a sequence of input tokens $u_t = (w^t_0, w^t_1, ..., w^t_L)$ and uses a (bi-directional) recurrent neural network to output a distributed user utterance representation $\mathbf{u}_t$ which is the final hidden state:

$$\mathbf{u}_t = \mathbf{h}^u_L = \mathrm{RNN}_\theta(u_t).$$

Wen et al. (2017b) defines this as a distributed intent representation which replaces the hand-coded dialogue act representation. In task-based end-to-end systems, the general sequence-to-sequence network is conditioned on two additional models which are essential for long-term dialogue modeling: 1) dialogue state tracking (DST) and 2) the knowledge base querying component. The summarized dialogue state is formed by concatenating three probability values: 1) a probability that the slot has not been mentioned until this turn; 2) a probability that the user does not care about this constraint; and 3) a sum of all other values for the given slot. This yields a 3-bin one-hot encoding $\mathbf{b}_{s_d,t}$ for each slot. The global dialogue/belief state vector is then formed by concatenating all slot-dependent vectors over all domains:

$$\mathbf{b}_t = \bigoplus_{d \in \mathcal{D}} \bigoplus_{s_d \in \mathcal{S}_d} \mathbf{b}_{s_d,t}, \tag{5.1}$$

where $\mathcal{S}$ is a set of of all slots in a domain. The current belief state $\mathbf{b}_t$ can be used to query the knowledge base. Based on the numbers of entities in the database that satisfy the current belief state, we form n-bin one-hot encodings for each domain $\mathbf{k}_d$. In all experiments, we use 6-bin encodings for 0, 1, 2, 3, 4, or more than 4 matches. All domain-dependent vectors are then concatenated into a global domain vector:

$$\mathbf{k}_t = \bigoplus_{d \in \mathcal{D}} \mathbf{k}_{d,t}. \tag{5.2}$$

5.3.2 Decision Making

In the next step, a policy vector is created to mimic the dialogue manager in the traditional modular approach. The intent vector $\mathbf{u}_t$, the belief state vector $\mathbf{b}_t$ and the knowledge database vector $\mathbf{k}_t$ are combined together and processed through a nonlinear layer:

$$\mathbf{a}_t = \tanh(\mathbf{W}_u \mathbf{u}_t + \mathbf{W}_k \mathbf{k}_t + \mathbf{W}_b \mathbf{b}_t). \tag{5.3}$$

This vector can be seen as a continuous version of a system act in the traditional modular approach summarizing the current state and action in a high-dimensional space.

5.3.3 Generation

The generation module uses a language model in the form of the recurrent neural network that outputs probabilities over the vocabulary set at each time step:

$$P(w_{j+1}|w_j, \mathbf{h}^m_{j-1}) = \text{softmax}(\text{RNN}(w_j, \mathbf{h}^m_{j-1})),$$

where w_j is the last output token and $\mathbf{h}^m_{t-1}$ is the hidden vector from the previous step. We condition a language generator through the action vector $\mathbf{a}_t$ by using it as the first hidden vector, i.e:

$$\mathbf{h}^m_0 = \mathbf{a}_t.$$

To begin the generation process we use a special token (signifying the beginning of a sentence, SOS) as w_0. The generation process stops when the network outputs a special token informing about the end of the sentence (EOS). The standard cross entropy is adopted as our objective function to train a language model:

$$L(\theta) = \sum_t \sum_j y_j^t \log p_j^t,$$

where y_j^t and p_j^t are output token targets and predictions respectively, at turn t of output step j. In our case every token is treated equally to make the model as simple and as general as possible.

Attention Mechanism

The signal from the user sentence can be amplified through the attention mechanism that enable focusing on the most relevant part of the utterance at each step. After each pass of the RNN over the user sentence the hidden vectors for each word are stored as:

$$\mathbf{h}^u = (\mathbf{h}_1^u, \mathbf{h}_2^u, ..., \mathbf{h}_L^u).$$

These are combined through a score function with the current hidden vector in the generation module $\mathbf{h}_{j-1}^m$:

$$a_{ij} = \text{score}(\mathbf{h}^u, \mathbf{h}_{j-1}^m) = \mathbf{v}^T \tanh(\mathbf{W}^T(\mathbf{h}_i^u \oplus \mathbf{h}_{j-1}^m) + \mathbf{b})$$

and passed through a softmax operator to produce the attention weights:

$$\alpha_{ij} = \frac{e^{a_{ij}}}{\sum_{i=1}^L e^{a_{ij}}}.$$

The current attentive hidden vector (the context vector) from the user sentence is created by weighting each hidden output with the attention weights:

$$\mathbf{h}_c^u = \sum_{i=1}^L \alpha_{ij} \mathbf{h}_i^u.$$

The context vector is concatenated with the previous generated embedded word

$$\mathbf{w}_{j-1}^{new} = \mathbf{h}_c^u \oplus \mathbf{w}_{j-1}$$

to output the prediction for the next word:

$$P(w_{j+1} | w_j, \mathbf{h}_{j-1}^m, \mathbf{h}^u) = \text{softmax}(\text{RNN}(\mathbf{w}_{j-1}^{new}, \mathbf{h}_{j-1}^m)).$$

5.4 Training Protocol

All models were trained for 20 full passes over the training dataset. The best set of hyper-parameters was found by a grid search, evaluating on the held-out validation subset of the dataset. We varied between the size of embedding vectors, hidden representation, $l2$-norm penalty and the size of the dictionary. The gradients are clipped by 5.0 throughout the whole training process. Three different recurrent architectures – standard RNN, its gated version (Chung et al., 2014) and bi-directional LSTMs (Graves et al., 2013) were analyzed with two different attention mechanisms (Bahdanau et al., 2014; Luong et al., 2015). However, no significant differences in performance

Evaluation Metrics Since often times the evaluation of a dialogue system without direct interaction with the real users can be misleading (Liu et al., 2016), three different automatic metrics are included to ensure the result is better interpreted. The fluency is measured via BLEU score (Papineni et al., 2002). The dialogue task completion is divided into two sub-tasks. First, the *Inform rate* evaluates whether the system has provided an appropriate entity, i.e. the entity that satisfies the constraints from the user's goal. If the dialogue spans across multiple domains, the score is positive only if in all domains entities satisfy the user's needs. If the Inform rate is positive, the *Success rate* evaluates whether the model answered all the requested attributes of provided entities.[2] Again, the metric is positive if the requested attributes were given in all domains. We consider both Inform and Success metrics since the former evaluates the ability to retrieve right entities and the latter evaluates the ability to reason and answer questions about entities. The best models on the validation set are chosen according to the following formula:

$$0.5 * \texttt{Inform} + 0.5 * \texttt{Success} + \texttt{BLEU}.$$

The formula enforces both task-completion and natural language realization.

5.4.1 Evaluation

Table 5.1 presents the results of various of model architectures trained on the new dataset with comparison to the Cam676 dataset (Table 4.3) (Wen et al., 2017b). The latter datasets consists of 676 dialogues in the CamRestaurant domain (Section 3.4) collected also in a Wizard-of-Oz fashion. The dialogues are significantly shorter without changes in the user goal throughout the dialogue. As expected, the model achieves an almost perfect score on

---

[2] The Success rate metric in the case of a single domain is the same as the one used in Chapter 3.

	Cam676		**MultiWOZ**	
	w/o attention	w/ attention	w/o attention	w/ attention
Inform (%)	99.17	99.58	71.29	71.33
Success (%)	75.08	73.75	60.29	60.96
BLEU	0.219	0.204	0.188	0.189

Table 5.1 Performance comparison of different model architectures using a corpus-based evaluation. The results are averaged over three models.

the Inform metric on the Cam676 dataset taking the advantage of an oracle belief state signal. However, even with the perfect dialogue state tracking of the user intent, the baseline models obtain almost 30 % lower score on the Inform metric on the new corpus. The addition of the attention improves the score on the Success metric on the new dataset by less than 1%. Nevertheless, as expected, the best model on MultiWOZ is still falling behind by a large margin in comparison to the results on the Cam676 corpus taking into account both Inform and Success metrics. As most of dialogues span over at least two domains, the model has to be much more effective in order to execute a successful dialogue.

To illustrate the complexity of the new policy task, Appendix A shows randomly picked dialogues from Cam676 and MultiWOZ where system's responses were generated by the trained models. The trivial policy of providing a restaurant and all necessary information achieves 100% performance since the user does not change her goal in Cam676. In MultiWOZ, the policy has to cope with changes of preferences, different domains and a booking procedure. Only then, the Inform metric will be positive in a given dialogue. Moreover, the BLEU score on the MultiWOZ is lower than the one reported on the Cam676 dataset. This is mainly caused by more diverse linguistic expressions observed in the MultiWOZ dataset.

5.5 MultiWOZ as a New Benchmark

The complexity and the rich linguistic variation in the collected MultiWOZ dataset makes it a great benchmark for a range of dialogue tasks. The usefulness in dialogue management task was shown in the previous section (Table 5.1). To further prove the potential of the MultiWOZ corpus, we report a benchmark result for two other standard tasks: dialogue state tracking, and dialogue-context-to-text generation. These results illustrate new challenges introduced by the MultiWOZ dataset for different dialogue modelling problems.

5.5.1 Dialogue State Tracking

A robust natural language understanding and dialogue state tracking is the first step towards building a good conversational system. Since multi-domain dialogue state tracking is still in its infancy and there are not many comparable approaches available (Rastogi et al., 2017), we instead report our state-of-the-art result on the restaurant subset of the MultiWOZ corpus as the reference baseline. The proposed method (Ramadan et al., 2018) exploits the semantic similarity between dialogue utterances and the ontology terms which allows the information to be shared across domains. Furthermore, the model parameters are independent of the ontology and belief states, therefore the number of the parameters does not increase with the size of the domain itself. The same model was trained on both the WOZ2.0 and the proposed MultiWOZ datasets, where the WOZ2.0 corpus consists of 1200 single domain dialogues in the restaurant domain (Table 4.3) (Mrkšić et al., 2017a). Although not directly comparable[3], Table 5.2 shows that the performance of the model is consecutively poorer on the new dataset for every slot compared to WOZ2.0. These results demonstrate how demanding is the new dataset as the conversations are richer and much longer.

Slot	**WOZ 2.0**	**MultiWOZ (restaurant)**
Overall accuracy	96.5	89.7
Joint goals	85.5	80.9

Table 5.2 The test set accuracies *overall* and for *joint goals* in the restaurant sub-domain.

5.5.2 Dialogue-Act-to-Text Generation

Natural language generation from a structured meaning representation (Bohus and Rudnicky, 2005; Oh and Rudnicky, 2000) has been a very popular research topic in the community, and the lack of data has been a long-standing block for the field to adopt more machine learning methods. Due to the additional annotation of the system acts, the MultiWOZ dataset serves as a new benchmark for studying natural language generation from a structured meaning representation. In order to verify the difficulty of the collected dataset for the language generation task, we compare it to the SFX dataset (Table 4.3), which consists of around 5k dialogue act and natural language sentence pairs. We trained the same Semantically Conditioned Long Short-term Memory network (SC-LSTM) proposed by Wen et al. (2015)

---

[3] The restaurant domain in MultiWOZ has more slots involving a booking procedure.

on both datasets and used the metrics as a proxy to estimate the difficulty of the two corpora. To make a fair comparison, we constrained our dataset to only the restaurant sub-domain which contains around 25k dialogue turns. To give more statistics about the two datasets: the SFX corpus has 9 different act types with 12 slots comparing to 12 acts and 14 slots in our corpus. The best model for both datasets was found through a grid search over a set of hyper-parameters such as the size of embeddings, learning rate, and number of LSTM layers. Table 5.3 presents the results on two metrics: BLEU score and slot error rate (SER). The significantly lower metrics on the MultiWOZ corpus showed that it is much more challenging than the SFX restaurant dataset. A smaller difference between BLEU scores than SER can be explained by the fact that BLEU is calculated based on 1-4 grams among all sentences. While the BLEU score can be high, the prediction does not match the key semantic terms, i.e slots. The complexity of input semantics is easier in SFX where only one act per example containing few slots. In MultiWOZ, however, more than 60% of the dialogue turns are composed of at least two system acts, and one-act can contain several slots.

Metric	**SFX**	**MultiWOZ (restaurant)**
SER (%)	0.46	4.378
BLEU	0.731	0.616

Table 5.3 The test set slot error rate (SER) and BLEU on the SFX dataset and the MultiWOZ restaurant subset.

5.6 Conclusion

The large-scale corpora collected and presented in Chapter 4 allows now to bootstrap the dialogue manager based on human-human dialogues. This chapter defines a set of benchmarks to illustrate the difficulty of collected conversations as well as to establish a gold standard for different approaches to task-oriented dialogue modelling. A multi-domain neural dialogue manager with multi-domain belief state, database pointer and attention component over decoded space was proposed. The model combines dialogue management with generation building upon single-domain neural end-to-end architectures. The proposed model was evaluated on the collected multi-domain dialogues. Compared to the single-domain set-up, the performance significantly drops as multi-domain dialogues are longer, often including two domains in one turn requiring better architectures that will allow to produce richer system outputs. Further, two other benchmarks: dialogue state tracking and generation are established. Again, modelling multiple domains and slots simultaneously poses significant challenges to the existent state-of-the-art approaches.

Chapter 6

Multi-Action and Slot Dialogue Agent for Managing Concurrent Actions

Building upon the baseline proposed in Chapter 5, this chapter proposes a novel reinforcement learning-based approach for training a **M**ulti-**A**ction and **S**lot **D**ialogue **A**gent (MASDA). In the first part, given the findings from the analysis of MultiWOZ, several limitations of current dialogue models are displayed. This includes handling concurrent actions present in a single turn and lack of interpretability, which consequently impedes the use of intermediate signals. In the second part of the chapter, we address these limitations with MASDA which explicitly models system acts and slots using intermediate signals, resulting in an improved task-based end-to-end framework. The model can also select concurrent actions in a single turn: this enriches the representation of generated responses. The evaluation of complex dialogues from the MultiWOZ dataset demonstrates that the proposed model outperforms previous end-to-end frameworks, while also offering improved scalability.

6.1 Motivation

Recent research puts increasingly more focus on end-to-end dialogue modeling (Eric et al., 2017; Lei et al., 2018; Liu et al., 2018; Wen et al., 2017a; Zhao et al., 2018). Relying on recurrent sequence-to-sequence modeling (Sutskever et al., 2014), the main idea is to propagate the learning signal through the entire dialogue system: this enables information sharing between all the constituent modules. However, the end-to-end models have their own limitations. They suffer from lack of long-term planning: this hinders their task completion effectiveness (Li et al., 2017a). Moreover, the collected corpus containing multi-domain multi-turn dialogues provide new challenges for efficient dialogue management within the end-to-end paradigm (Chapter 4). In particular, multi-action and multi-slot response generation are handled through hand-crafted and non-scalable summary action spaces in current reinforcement learning (RL)-based approaches to dialogue management (Gašić et al., 2010). Several recent works tackle this problem with learning underlying dialogue intent through unsupervised modelling (Wen et al., 2017a; Zhao and Eskenazi, 2018; Zhao et al., 2018). These models rely on the neural variational inference frameworks (Mnih and Gregor, 2014) to handle complex dialogue acts. However, this modeling paradigm currently lacks clear-cut boundaries between different discrete latent encodings as their homogeneity results are not yet satisfactory (Zhao et al., 2018). As a consequence, these models currently cannot fully exploit intermediate signals (e.g., dialogue annotations) if such signals are available. Contrary to the case of open-domain domains, task-oriented dialogues often can be described in the district space of actions and benefit from supervision.

In this chapter, we propose a novel end-to-end learning model dubbed MASDA: **M**ulti **A**ction and **S**lot **D**ialogue **A**gent. The proposed model utilises the explicit signal originating from the intermediate annotations of system acts and slots (Traum, 1999) to inform the dialogue management module. The inclusion of the explicit additional supervised signal results in fully interpretable actions taken by the model. It also enables RL-based fine-tuning of the management module. In summary, the main contributions of this chapter are as follows. First, we propose and validate a scalable RL-based framework for managing concurrent actions. Such actions are ubiquitous in multi-domain conversations but have been typically discarded by previous end-to-end approaches. Second, we demonstrate that intermediate dialogue annotations for acts and slots improve task-completion effectiveness for end-to-end dialogue systems. Finally, we combine both concurrent action modeling and slot information: this yields our final MASDA model. The model is pretrained using the supervised learning signal, and further autonomously fine-tuned with RL. The results indicate that managing concurrent actions and leveraging explicit intermediate signals in the form of dialogue act and slot annotations improve task-completion metrics. With the scalable RL-based fine-tuning process, our model pushes the policy to new state-of-the-art results on the MultiWOZ dataset.

6.2 Complex Multi-Domain Dialogues

A prominent challenge introduced in multi-domain dialogues concerns the observation that a large number of dialogue's turns, in fact, comprises more than one dialogue act, as illustrated in Figure 4.10 from Section 5. The richness of system utterances is exemplified by the fact that almost 50% of turns contain more than one dialogue act. What is more, the total number of intent combinations in MultiWOZ is 281. An often recurring combination of dialogue acts

(Section 2.5) concerns *Inform* and *Request* acts. For example, the system utterance: *"I have found 4 hotels satisfying your criteria. Do you have any preference for the area?"* can be labelled as:

```
Inform(domain=hotel,price=moderate,entities=4)
Request(domain=hotel,area)
```

In the example, *Inform* and *Request* are two distinct dialogue acts found in a single system utterance. Thanks to the more complex utterances, wizards could serve the users faster by providing more information (*Inform*) while leading the dialog flow (*Request*). However, the current state-of-the-art RL-based policy models do not accommodate for operating with concurrent actions (Fatemi et al., 2016; Gašić et al., 2010; Wen et al., 2017a). The standard approach takes its next decision based on probabilities parametrised through a softmax gate. In single-domain settings comprising a small number of actions, it is possible to enlist and model all possible dialogue act combinations. However, this strategy does not scale well: it is not viable for more complex conversations covering multiple acts and domains (Casanueva et al., 2018).

Moreover, a very desired characteristic of machine learning models is the interpretability (Doshi-Velez and Kim, 2017). Interpretable system actions allow human to understand the behaviour of a dialog system and better interpret the system intentions (Wen et al., 2017a). Secondly, the interpretability of system actions eases up debugging the cause of errors and guide further data collections. Also, if some system actions have higher business costs, additional rule-based constraints can be added to ensure the robustness of the policy. Current unsupervised modeling paradigms lack clear-cut boundaries between different discrete latent encodings (Wen et al., 2017a; Zhao et al., 2018). However, we can take advantage of the fact that MultiWOZ provides intermediate labels for actions and slots that can be used in a supervised pre-training phase.

Table 6.1 lists all dialogue acts present in MultiWOZ. There are 14 different dialogue acts in total. The *No annotation* act is assigned to turns that the human annotators found difficult to categorise. Out of the 14 acts, 4 acts are *system-specific*: they typically depends on crowdsourced workers communication style. These actions are more challenging to predict and model, as they are an artefact of the data collection procedure; they depend on the conversation style of the person having run the system side during data collection. In what follows, we test the policy module with two action sets: **1) Full** contains all 14 actions, while **2) Constrained** is reduced only to 10 actions that are not system-specific, see Table 6.1.

Dialogue act	**System-specific?**
Request	✗
Inform	✗
Request more	✓
Recommend	✗
Select	✗
No entities available	✗
Goodbye	✓
You are welcome	✓
Booking proposal	✗
Book	✗
Booking confirmation	✗
Booking rejection	✗
Greet	✓
No annotations	✗

Table 6.1 The list of all 14 labelled actions occurring in the MultiWOZ corpus. The system specific flag signifies whether the action typically depends on the turker communication style.

6.3 Managing Concurrent Actions

We now describe the proposed policy model for managing concurrent actions, along with its training procedure spanning both supervised and reinforcement learning phases. Figure 6.1 presents the full architecture of the system. The baseline framework is built upon the encoding of the state as defined in Section 5.4.

6.3.1 Explicit Policy Module

Sequence-to-sequence architectures (Sutskever et al., 2014; Vinyals and Le, 2015) are known to be inefficient at retaining long-term planning of the conversation even with attention-based approaches (Parthasarathi et al., 2020; Serban et al., 2017; Wei et al., 2018). One way to mitigate this limitation is by supplementing the architectures with an explicit policy module that encodes predefined set of actions. This was empirically proven to improve task-based dialogue systems (Wen et al., 2017a; Zhao et al., 2018). The policy module is typically instantiated as a feed-forward neural network that takes the vector $\mathbf{x}_t$ from Equation (5.3) as its input (see Section 5.3 again). Comparing to previous works, we have a feed-forward network in the policy module with a sigmoid in the last layer that allows to predict separately

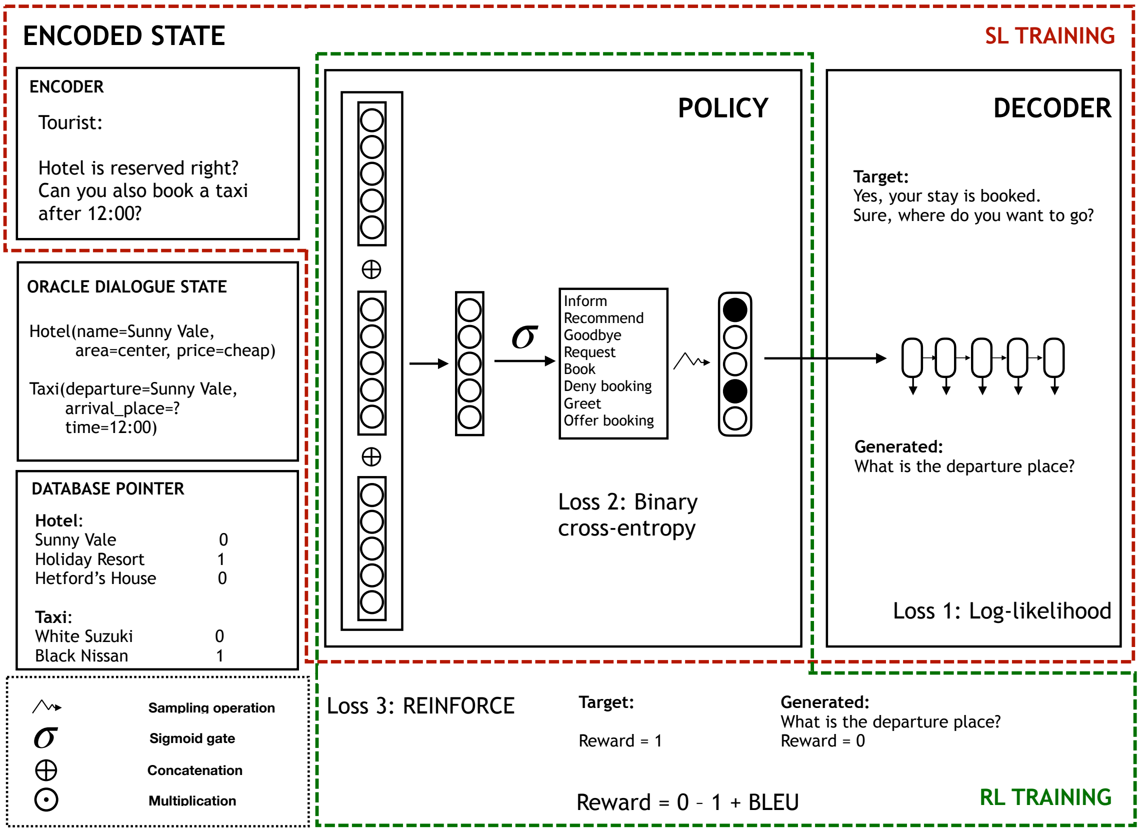


Fig. 6.1 High-level architecture of the multi-domain management and response generator with three losses.

the probability of choosing each of possible actions. This allows us to have more than just one action per system sentence. According to probabilities we can sample from the distribution or take actions that probability is higher than 0.5. This decision vector is then passed further to the decoder.

The feed forward net consists of three non-linear layers with tanh activations: $\mathbf{z}_t = \mathrm{MLP}(\mathbf{x}_t)$. The probabilities over the action space are then defined as:

$$\pi(\mathbf{a}_t|\mathbf{x}_t) = \phi(\mathbf{z}_t), \tag{6.1}$$

where ϕ denotes operation of multi-dimensional *sigmoid* or *softmax*. In the case of sigmoid activitation, following Harmer et al. (2018), we assume that each action is conditionally independent given the state representation $\mathbf{x}_t$, that is:

$$\pi_\theta(\mathbf{a}_t|\mathbf{x}_t) = \prod_{n=1}^{N} \pi(a_t^n|\mathbf{x}_t), \tag{6.2}$$

where a_t^n is the n-th action from the action set $\mathcal{A}$. We note here that this simplification is not justified by the data, but by the need to simplify the mathematical modeling of the framework. Dialogue actions are clearly related; actions such as *Greet* and *Goodbye* should not occur together. In another example, issuing booking action before fixing the correct entity should be prevented. Nevertheless, we hope that the expressiveness of the policy network provides enough context to allow the sigmoid layer to model the relations between probabilities correctly. This assumption allows us to treat each action as a Bernoulli random variable leading to:

$$\pi_\theta(\mathbf{a}_t|\mathbf{x}_t) = \prod_{n=1}^{N} \left(a_t^n z_t^n + (\mathbb{1} - a_t^n)(\mathbb{1} - z_t^n)\right). \tag{6.3}$$

Now, both activations give output that can be interpreted as probabilities and thus we can sample the discrete action (or actions):

$$\mathbf{a}_t^s \sim \pi(\mathbf{a}_t|\mathbf{x}_t). \tag{6.4}$$

Let us denote by $\mathbf{W}$ and $\mathbf{b}$ any weight matrix and bias, respectively. Following Wen et al. (2017a), we force the mixing of information encoded in the state representation ($\mathbf{x}_t$) with the chosen action vector through their direct multiplication. This provides us with a context vector used in the response generation part:

$$\mathbf{c}_t = (\mathbf{W}_1^T \cdot \mathbf{a}_t + \mathbf{b}_1) \odot (\mathbf{W}_2^T \cdot \mathbf{x}_t + \mathbf{b}_2). \tag{6.5}$$

The first hidden vector used in the response decoder is formed by concatenating $\mathbf{c}_t$ with the action vector to reinforce the signal and then transforming it through a non-linear layer again:

$$\mathbf{h}_0 = \tanh(\mathbf{W}_3^T \cdot (\mathbf{a}_t \oplus \mathbf{c}_t) + \mathbf{b}_3). \tag{6.6}$$

6.3.2 Sigmoid vs. Softmax

The set of all possible actions $\mathcal{A} = \{a_1, \ldots, a_N\}$ consists of N individual actions which is any subset of the action set from Table 6.1. Standard RL approaches restrict the subset to contain only one action per time step (Sutton and Barto, 1999) through stochastic policy $\pi : \mathcal{X} \rightarrow \mathcal{A}$ where $\mathcal{X}$ refers to the input space. To enable managing concurrent actions (e.g., necessary to model complex MultiWOZ dialogues), the output space has to grow exponentially. If all possible action subsets are allowed, the action space becomes a one-hot encoding of 2^N possible outputs. Even for a reasonably small action space from Table 6.1 (14 actions), the number of possible outputs is $16,384$.

One solution to mitigate the output space explosion issue is to replace softmax with the *sigmoid* output in Equation (6.1). The sigmoid activation scales linearly with the number of actions: this property enables learning more complex conversations, as long as the parametrised policy has enough learning capacity. Allowing a more complex latent action space was proven beneficial for unsupervised learning settings in prior work (Zhao et al., 2017, 2018). In this work, we hypothesise that it is also beneficial in our scenario where the explicit learning signal is provided through intermediate labels.

6.3.3 Pretraining with Supervised Signal

Let us denote by θ all parameters of the model from all three modules. The main loss function is the cross-entropy between the prediction values of generated words and target (gold) sentences:

$$L_1(\theta) = \sum_d \sum_t \sum_j y_j^t \log p_j^t. \tag{6.7}$$

The summation goes over all training dialogues (d), turns of each dialogue (t) and each word (j). p_j^t is the probability of the output token j, and y_j^t is the indicator variable for each target token.

Our model augments $L_1(\theta)$ with another intermediate loss function for explicit action prediction. This cross-entropy-based loss is as follows:

$$L_2(\theta) = \sum_d \sum_t (\mathbb{1} - \mathbf{a}_t)(\mathbb{1} - \log \mathbf{p}_a^t) + \mathbf{a}_t \log \mathbf{p}_a^t. \tag{6.8}$$

It predicts the correct actions at dialogue d and for turn t; $\mathbf{a}_t$ is the oracle action vector.

We do not sample the action vector during training but use the oracle action vector instead. This leads to the separation of gradients for all the modules. Therefore, the two losses from Equation (6.7) and Equation 6.8 are simply summed together in the supervised learning (SL) phase without any further importance weighting.

6.3.4 RL-Based Fine-Tuning

The losses $L_1(\theta)$ and $L_2(\theta)$ are not directly related to the actual dialogue evaluation metric. This means that the supervised learning phase does not effectively exploit the full potential of training data at hand. Here, the RL fine-tuning phase can be used to add the signal concerning the success of the current dialogue policy. This approach has been previously validated by, e.g., Wen et al. (2017a) and Lei et al. (2018). We denote by r_t a total sum of rewards until turn t. A standard REINFORCE algorithm (Williams, 1992) can be applied (Equation 2.19) when the policy module samples with the softmax activation resulting in the third loss:

$$L_3(\theta) = -\frac{1}{D}\sum_{d=1}^{D}\frac{1}{T}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{x}_t) r_t, \tag{6.9}$$

where T is the length of the dialogue d and D is number of dialogues in the batch.

In the case of the sampling multiple actions, we substitute the softmax with sigmoid. $L_3(\theta)$ now gets reformulated to:

$$\begin{aligned} L_3(\theta) &= -\frac{1}{D}\sum_{d=1}^{D}\frac{1}{T}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{x}_t) r_t \\ &= -\frac{1}{D}\sum_{d=1}^{D}\frac{1}{T}\sum_{t=1}^{T}\nabla_\theta \sum_{n=1}^{N} [a_t^n \log(z_t^n) \\ &\quad + (\mathbb{1} - a_t^n)\log(\mathbb{1} - z_t^n)\,]\, r_t. \end{aligned} \tag{6.10}$$

We refer the reader to Figure 6.1 again for the overview of the model structure along with the three losses from Equation (6.7), Equation (6.8), and Equation (6.10) used to train the model.

6.4 Multi-Action and Slot Dialogue Agent

The framework proposed in Section 6.3 can be naturally extended to also include explicit modeling of slot values for each domain. The necessity of providing multiple values in one turn is even more apparent. The sigmoid approach can be reused here as well. The problem can be formulated as a hierarchical multi-label classification: it can be tackled by, e.g., a recent hierarchical classification model of Wehrmann et al. (2017).

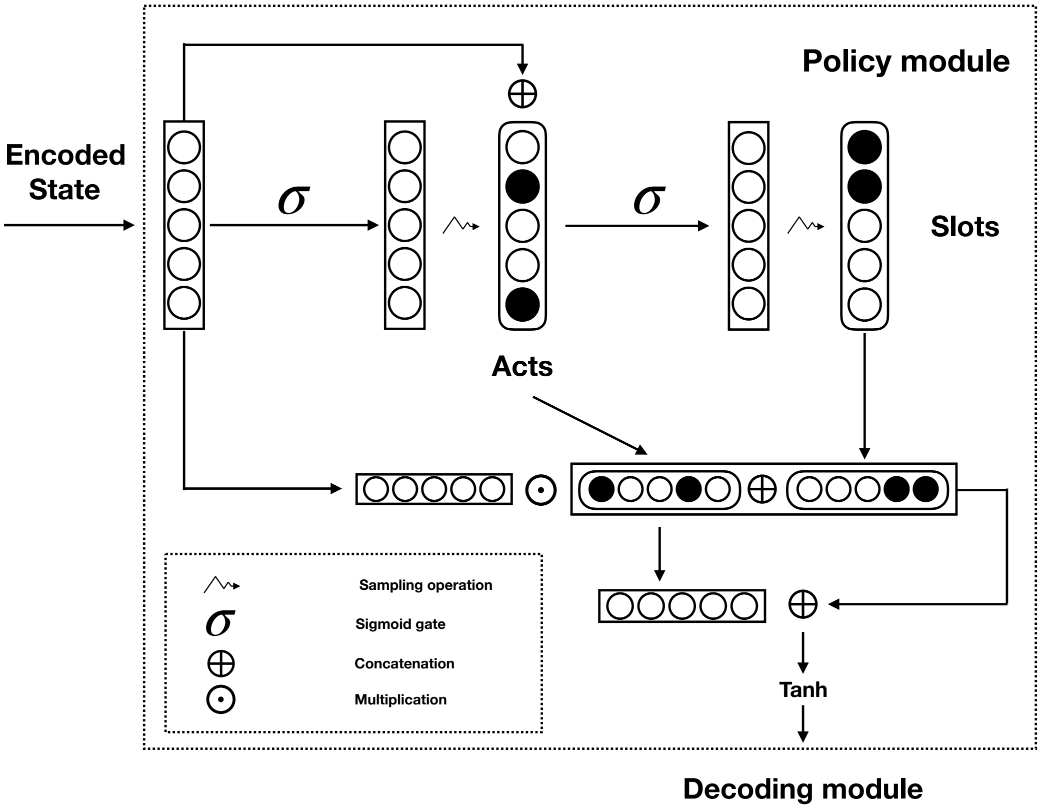


Fig. 6.2 High-level architecture of the policy module in the proposed MASDA model described in Section 6.4.

6.4.1 Joint Modeling of Actions and Slots

As in Section 6.3, the input to the policy module is again the encoded state $\mathbf{x}_t$ that consists of information about the current user utterance, knowledge base information and the oracle belief state at the current turn, see Equation (5.3). The probability for each action is first computed using a sigmoid gate σ:

$$\mathbf{p}_t^a = \sigma(\mathbf{W}_1^T \cdot \tanh(\mathbf{x}_t) + \mathbf{b}_1). \tag{6.11}$$

Again, at test time this allows us to sample actions from each dimension independently: $a_t^i \sim p_t^i$. On the other hand, during training we pass the labelled vectors $\mathbf{a}_t$. The action vector $\mathbf{a}_t$ is concatenated with the encoded state vector $\mathbf{x}_t$ to obtain predictions for slot values as follows:

$$\mathbf{p}_t^s = \sigma(\mathbf{W}_2^T \cdot (\mathbf{a}_t \oplus \mathbf{x}_t) + \mathbf{b}_2). \tag{6.12}$$

The sampling procedure used here to obtain the slot vector $\mathbf{s}_t$ is exactly the same as for the action vector. The slot vectors and the actions vectors are concatenated together and interact with the encoded state $\mathbf{x}_t$ as follows:

$$\mathbf{z}_t = (\mathbf{W}_3^T \cdot (\mathbf{a}_t \oplus \mathbf{s}_t)) \cdot (\mathbf{W}_4^T \mathbf{x}_t). \tag{6.13}$$

In order to reinforce the system decision, the reinforcement signal is further combined with the action and slot vectors:

$$\mathbf{h}_0 = \tanh(\mathbf{W}_5(\mathbf{z}_t \oplus \mathbf{s}_t \oplus \mathbf{a}_t) + \mathbf{b}_5). \tag{6.14}$$

The output $\mathbf{h}_0$ of Equation (6.14) is then passed as the initial hidden state to the decoder in the response generator. Figure 6.2 presents the final architecture of the multi-action and slot policy module in the proposed MASDA model. It is important to note that with the current MASDA design it is not possible to provide to the decoder explicit information concerning the relationship between chosen actions and slots. We assume that the generator given the context contains all information for combining correct slots with actions.

6.4.2 Training Procedure

Since we have extended the model from Section 6.3 to now also model slot probabilities, we augment the training procedure with another loss function to take this into account. It minimises the cross-entropy between predicted probabilities of slots and ground-truth labels:

$$L_4(\theta) = \sum_d \sum_t (\mathbb{1} - \mathbf{s}_t)(\mathbb{1} - \log \mathbf{p}_s^t) + \mathbf{s}_t \log \mathbf{p}_s^t, \tag{6.15}$$

where d is a dialogue in the batch, t is a turn in the dialogue d, and $\mathbf{s}_t$ is the oracle slot vector. The RL fine-tuning phase is similar to the one from the previous section. We again only optimise with respect to the parameters related to action selection, leaving all other parameters fixed.

In the supervised learning phase, the loss functions from Equation (6.7), Equation (6.8), and Equation (6.15) are combined. The intermediate act and slot labels were passed to the generator instead of passing probabilities: this makes the encoder and policy modules disconnected from the decoder module. The loss function from Equation (6.15) is used to connect the policy module and the decoder module. During the RL fine-tuning phase we keep all parameters fixed except the parameters of the policy module to avoid overfitting (Lei et al., 2018; Wen et al., 2017a). However, contrary to the prior work, we have not observed any deterioration in performance after several epochs; with the MASDA model it simply stabilises at the highest peak.

6.5 Experiments and Results

The proposed methods are evaluated on the MultiWOZ corpus employing the metrics from Section 5.4. Given the poor performance of the initial DST module on this dataset (Ramadan et al., 2018), we use the oracle belief state predictions (Section 2.4), see Figure 6.1. The selected design currently prevents us to test the proposed methodology with human subjects as the model cannot obtain belief state predictions at the inference time. However, at the same time it facilitates the analysis of the results, as the language understanding/DST module is deliberately separated from the policy and the generation modules. As the focus of this work is on the two latter modules, by operating with exactly the same DST module with all models in evaluation, we can attribute all differences in the results to the changes in the design of the policy and generation modules.

6.5.1 Experimental Setup

Baselines. The first baseline is a neural response generation model with an *oracle* belief state obtained from the wizard annotations while the policy module is represented by a simple feed-forward network (see Chapter 5). As mentioned, using the oracle belief state allows us to obtain a clear benchmark where the performance of dialogue management and response generation is completely independent of the understanding module.

The second baseline can be formulated as "MASDA *with softmax*": it refers to the same policy module proposed in the previous sections, but using softmax instead of sigmoid in the last layer. In this case, the policy module predicts one-hot encoding over all possible actions. This baseline demonstrates the benefits of utilizing sigmoid.

Training Setup. The encoder and the decoder are single-directional GRUs (Cho et al., 2014) as we did not observe any improvement from using bidirectional recurrence. The embedding vectors are of size 50 while hidden vectors for both the decoder and the encoder are of size 150. The final MASDA model with the sigmoid activation consists of $481,281$ parameters, while the baseline model contains $323,300$ parameters. The softmax version of the MASDA model consists of $619,312$ parameters (with the *Constrained* action set of 10 actions, see Table 6.1) and $11,134,768$ parameters (with the *Full* set of 14 actions). All other hyper-parameters are presented in the Appendix B.

All models are trained with 15 full iterations over the training set in both the SL and RL training phases. All results are averaged over 10 different seeds. For the RL fine-tuning phase, the best epochs on the validation set have been chosen for further training. The reward function is defined as in (Wen et al., 2017a) and the fine-tuning RL phase is constrained to update only the policy module. This steers the model to optimise the policy module solely, while keeping the remaining components (i.e., language understanding and generation) fixed throughout the RL phase.

6.5.2 Supervised Learning Phase

The first set of experiments works with the *Constrained* set of 10 actions related to task completion (see Table 6.1 again for the two sets of actions). The results reported in Table 6.2 reveal two important findings. First, the explicit policy modeling gives improvements in both *Inform* and *Success* metrics. Both with softmax and sigmoid the model with SL optimisation improves over the original baseline on both evaluation measures. Second, the difference in performance between the two model variants (sigmoid vs. softmax) is not substantial. However, the sigmoid-based model requires only 10 output values, while the softmax-based model relies on $1,024$ output values.

	Baseline	Sigmoid	Softmax
Inform (%)	78.71	**82.49**	82.19
Success (%)	65.21	66.95	**68.11**
BLEU (%)	17.70	**18.81**	18.79

Table 6.2 The performance of the baseline model and two considered policy modules on the *Constrained* action set (10 actions) relying only on the SL phase.

This design difference is amplified when we consider all 14 actions from the *Full* set. The softmax layer now needs $16,384$ output values, while sigmoid needs only 14. As expected, Table 6.3 now shows that the sigmoid-based model further improves, while the softmax-based model struggles with generalization and obtains worse results.

6.5.3 Multi-Action Reinforcement Learning

As validated in Section 6.5.3, the explicit intermediate signal for acts improves the *Inform* metric by $\approx 6\%$ and the *Success* rate by $\approx 2\%$ when sigmoid is used. We now fix the best models based on the evaluation from Section 6.5.3, and proceed to the RL fine-tuning phase.

	Sigmoid	Softmax
Inform (%)	**84.15**	83.00
Success (%)	**67.42**	34.34
BLEU (%)	**17.76**	9.81

Table 6.3 Performance on the *Full* MultiWOZ action set (14 actions); SL phase only.

	Sigmoid (*Constrained*)	Sigmoid (*Full*)	Softmax (*Constrained*)	Softmax (*Full*)
Inform (%)	84.52	**85.79**	82.10	83.08
Success (%)	67.57	**67.63**	48.18	34.34
BLEU (%)	**17.36**	15.60	14.81	9.86

Table 6.4 Results using the RL fine-tuning phase based on the loss function from Equation (6.15).

Table 6.4 shows the main results after applying the RL fine-tuning step. Only the sigmoid-based variant allows for additional improvements of 2% on the *Inform* metric while not deteriorating the BLEU score significantly. The softmax variant displays low *Success* scores with both action sets. We have also observed that the results depend on the way the action vector **a** is chosen. The best results are reported with a mixed setup where the actions are sampled during training, while a threshold of value 0.5 is imposed at test time.[1]

6.5.4 Evaluating the Full MASDA Model

After establishing the improved performance with a sigmoid activation function, we can test the performance of the model equipped with all supervision signal including both intents and slots. For the experiments with the full MASDA model we use the full set of 14 actions. The additional signal leveraging explicit slot labels has now been added during the SL phase. As expected, Table 6.5 shows that the model exploiting both act and slot information compares favorably to the previous model which utilises only dialogue act information (reported in Table 6.3). We observe on par performance of the full MASDA model on the *Inform* metric, while at the same time the model significantly improves on the *Success* evaluation.

Furthermore, the RL fine-tuning phase now contributes even more than before. Table 6.5 reports substantial improvements with the RL phase, and peak scores overall on both *Inform* and *Success* metrics. The full sigmoid-based MASDA model with the RL fine-tuning phase

---

[1] Doing only thresholding or only sampling in both training and test yields subpar performance for both sigmoid and softmax variants across all experimental runs.

	Sigmoid SL	Sigmoid RL
Inform (%)	83.97	**88.34**
Success (%)	71.44	**75.41**
BLEU (%)	**16.78**	15.95

Table 6.5 Results of the full MASDA model which leverages both dialogue act and slot information.

leads to improvements of $\approx 10\%$ on *Inform* and *Success* metrics over the baseline model without a strong decrease of the BLEU score. We hypothesise that the MASDA model avoids squeezing out all information of rich dialogue acts (including slots) into one action and passes more enriched context vector to the decoding module. In summary, the gradual improvements reported in Tables 6.2-6.5 have revealed the importance of all integral MASDA components: 1) using intermediate signals and managing concurrent actions, 2) using sigmoid instead of softmax to improve scalability, 3) joint modeling of acts and slots, and 4) the new RL fine-tuning framework for managing concurrent acts after the SL pretraining phase.

6.6 Conclusions

Multi-domain large dialogue corpora now allow training structurally more complex dialogue agents. However, the new challenges with longer, more complex, and semantically richer multi-domain conversations also require adding explicit inductive bias to the decision-making module. We have presented a multi-action and slot dialogue agent (MASDA) that can handle concurrent actions frequently found in multi-domain dialogues, and additionally integrate available turn-level slot information. By using intermediate action and slot labels from annotated dialogues, our model learns to perform a more efficient dialogue policy that leads to higher task-completion rates. The new framework with the concurrent reinforcement learning approach further improves the policy module while preserving its scalability.

Future work could lead towards a full end-to-end model, and replace the oracle belief state used here with the learned belief state module. This will allow for user-centered evaluation and using the RL fine-tuning paradigm through interactions with human subjects.

Chapter 7

Conclusion

Multi-domain dialogues introduce a new level of complexity in all modules in the SDS pipeline. This thesis focused on the challenges of dialogue management where the limitations of hand-crafted rules in modelling the agent's policy become even more apparent than in the single domain. Specifically, two challenges were tackled in data-driven multi-domain policy learning: sample efficiency in new domains and grounding the policy in human-human conversations.

Reinforcement learning casts a policy design in a data-driven framework, seeking an optimal policy that yields the highest rewards and it alleviates the need for writing hand-crafted logic. Although the deep RL paradigm allows tackling complicated domains, sample efficiency becomes a major obstacle in learning from direct interactions with real users. This thesis shows that the agent can be enriched with uncertainty estimates over the parameters that lead to an informed policy exploration. The agent can explore the environment while being guided by the uncertainty rather than by a random choice, thereby increasing the learning speed. Moreover, the learnt policies become more robust in adversarial environments with high signal noise due to ASR or NLU errors.

However, learning of the agent's policy directly from real-world interactions is often not possible (industrial requirements) or available (the introduction of a new domain). The thesis presents an overview of the currently available dialogue corpora and data collection paradigms that could help to bootstrap initial policies. The analysis highlighted the great necessity of large-scale human-human dialogue datasets. This has been addressed with a proposal of a crowdsourced collection paradigm. The paradigm was validated by a collection of the largest task-oriented dialogue dataset: MultiWOZ. The collected corpus showcases the complexity of long-term multi-domain conversations where the ontology spans multiple slots and values while the user goal involves multiple sub-goals. The thesis proposes three benchmarks using state-of-the-art architectures to demonstrate new challenges.

Finally, the thesis proposes a new paradigm for multi-domain dialogue management. Thanks to the collected corpus, the dialogue manager's policy can be bootstrapped with human-human conversations without the need for modelling sub-policies separately. Rather, the agent learns a general paradigm of conversations observed across domains. The thesis proposes a MASDA model that combines the pre-training phase with human-human conversations and the fine-tuning stage using the RL paradigm to further improve the policy. The naturalness of conversations exemplifies the need for handling multiple actions present in a single turn. These limitations were addressed by explicitly modelling system acts and slots using intermediate signals, resulting in an improved task-based end-to-end framework. The model can also select concurrent actions in a single turn: this enriches the representation of the generated responses. The evaluation of the complex dialogues from the MultiWOZ dataset demonstrates that the proposed model outperforms previous end-to-end frameworks while also offering improved scalability.

7.1 Summary of Contributions

The presented research on data-driven dialogue management powered by deep reinforcement learning offers several interesting takeaways, which are summarised in the following sections.

Bayesian Deep Reinforcement Learning Improves Policy Speed and Robustness. Modelling uncertainty estimates with neural networks allows for informed exploration, as the action taken by the agent is not randomly chosen. Rather, exploration is coupled with the uncertainty of the model which improves the sample efficiency (Lipton et al., 2018; Osband et al., 2016). This helps to more rapidly learn dialogue policies in new domains. The application of several families of Bayesian networks shows that deep RL can fully match non-parametric models while greatly reducing the computational complexity. First, the results confirm that exploration driven by the uncertainty rather than ϵ-greedy-based methods improves learning efficiency in the early stages. What is more, the policies are less prone to overfitting, which makes them more robust against the noise errors that are often experienced in spoken dialogue systems. Interestingly, robustness transfers across different noise levels.

Deep Reinforcement Learning for Low-data Large-ontologies Regimes. The training with the Bayes-by-Backprop approach shows the potential to outperform the state-of-the-art in policy optimisation – GP-SARSA – particularly in more challenging domains, while greatly outperforming the latter in computational complexity. The GP-SARSA model has been a long-standing paradigm in dialogue management (Casanueva et al., 2017; Gašić et al., 2011; Su et al., 2017). However, deep-RL coupled with Bayesian exploration offers the scalability greatly needed in the case of growing size of domains, ontologies, and active users of deployed conversational assistants.

Data as an Important Bottleneck in Dialogue Modelling. This thesis presents a novel overview of available dialogue corpora and different data collection paradigms. Contrary to other NLP problems, the dialogue domain is greatly under-resourced. The available datasets are usually constrained in one or more dimensions such as missing proper annotations, only available in a limited capacity, lacking multi-domain use cases, or having a negligible linguistic variability (Asri et al., 2017; Henderson et al., 2014b; Wen et al., 2017b). To address these requirements, we collected the MultiWOZ corpus spanning multiple task-oriented domains. The new challenges introduced by the corpus have been already adopted by the community and have been actively tackled across different SDS modules. As a result, the latest 8th edition of the Dialogue System Technology Challenge introduced an end-to-end task completion track based on MultiWOZ (Kim et al., 2019).

Large-Scale Collection of Domain-oriented Conversations is Feasible. This thesis introduced the largest task-oriented dialogue corpus available to the research community. The paradigm allows for the collection of domain-specific conversations. The collection is performed entirely through crowdsourcing without the need for highly specialized and costly annotators. The Wizard-of-Oz set-up was designed to provide an easy-to-operate system interface for the wizards and easy-to-follow goals for the users. The small cognitive load of the required tasks enables for the collection of long-term conversations that span multiple domains. After the release of MultiWOZ, both academic and industrial research groups has built upon our proposed paradigm, leading to the introduction of several new dialogue corpora (Byrne et al., 2019; Lee et al., 2019; Peskov et al., 2019; Rastogi et al., 2019). The multitude of new corpora validates the crowdsourcing paradigm that allows for the data acquisition required to bootstrap dialogue agents in a time and cost-effective manner.

Policy can be Bootstrapped with Real-world Dialogues and Fine-tuned with Reinforcement Learning. Multi-domain dialogue management has been typically modelled through a set of isolated policies sharing a general dialogue manager that guides a conversation's long-term goal (Cuayáhuitl, 2009; Gašić et al., 2015; Lison, 2011; Wang et al., 2014). This thesis proposes a unified framework where the policy module is shared across many domains. This set-up naturally promotes knowledge sharing across domains and tasks without explicitly defining the structure and greatly reducing modelling efforts. The proposed model empirically validates the challenges introduced by MultiWOZ. Next, the multi-domain neural dialogue manager is cast back into the reinforcement learning framework by the incorporation of the explicit intermediate signal of dialogue acts. The explicit action space allows for the interpretation of the underlying agent's actions against the natural realization. The model seamlessly integrates real human-human conversations with continual learning through interaction with users (Su et al., 2016).

Dialogue Agent can Learn with Concurrent Actions. The new corpus shows the importance of combining multiple actions in one turn. Contrary to other domains, the agent often operates on a limited number of turns, and each decision has a fundamental impact on the user's response. This thesis introduces a new multi-action loss into the RL paradigm (Harmer et al., 2018). The loss allows for training an agent that can issue multiple actions at once using both collected conversations and can fine-tune with RL afterwards. The RL fine-tuning pushes the model's performance, achieving a new state-of-the-art, as the agent has a higher degree of freedom in combining multiple actions. Moreover, the loss keeps the quality of the generation intact, contrary to standard RL solutions (Zhao et al., 2019). Further improvements are realized through the addition of intermediate signals for slots where the agent explicitly models what parts of the ontology are currently being discussed.

7.2 Challenges and Future Work

The introduction of large-scale task-oriented corpora with recent developments in transfer learning opens an exciting new chapter in modelling dialogue systems. This thesis concludes with open questions that could be addressed next.

Transfer Learning for Dialogue Systems. Transfer learning caused a fundamental paradigm shift in training a variety of NLP models (Howard and Ruder, 2018; Radford et al., 2018, 2019). This shift applies to both modelling paradigms and the collection of new resources. First, the pre-trained large transformer-based architectures (Vaswani et al., 2017) require either a simple fine-tuning step or the addition of the final layer on top of the existing structure. Moreover, acquiring an understanding of the general characteristics of the natural language, thanks to pre-training on large corpora, allows shifting the burden of learning them away from task-oriented corpora. This change moves the stress from collecting a large amount of data in a particular domain to high-quality challenging datasets that allow for robust fine-tuning in a particular problem (Nie et al., 2019). Task-oriented dialogue modelling inherently operates on low data regime, and the application of large pre-trained sentence encoders to the task-oriented dialogue domain seems natural. The first applications in the task-oriented domain involve both retrieval (Henderson et al., 2019; Humeau et al., 2019) and generative structures (Budzianowski and Vulić, 2019; Mehri et al., 2019; Wu et al., 2019b). Pre-training on large conversational datasets like Reddit with a response-selection loss is beneficial in a highly constrained domain like customer support (Humeau et al., 2019) or restaurant browsing (Henderson et al., 2019). Furthermore, even the sole pre-training on MultiWOZ improves performance, convergence, or domain generalizability for different down-stream tasks like dialogue act prediction or response selection (Mehri et al., 2019). Budzianowski and Vulić (2019) proposed a task-oriented dialogue model that operates solely on text input. This allows for directly fine-tuning large pre-trained language models, for example, GPT-2 (Radford et al., 2019). Wu et al. (2019b) modelled different speakers in alternating order with two pre-trained language models and achieved a superior task-completion performance on MultiWOZ. All these findings show the potential of unifying generation and retrieval losses for better pre-training results and direct application to task-oriented domains. The potential pre-training could use higher-quality data for a generative loss[1] with pure conversational data for the response-selection pre-training.

Multilingual Conversations. A significant portion of research in dialogue systems, and natural language processing in general, has an English-centric bias (Bender, 2019). This bias also applies to available dialogue corpora. The sole focus on the English language in building conversational AI cuts out a large pool of potential users and customers who prefer their native language to English. What is more, the performance of state-of-the-art architectures built on top of English resources drops by a large margin when faced with new languages (Mrkšić et al., 2017b; Schuster et al., 2019). The collection pipeline proposed in this thesis naturally allows for collecting new dialogues with crowd-workers from different countries. The original corpus could be also translated into other languages to facilitate a direct comparison across a variety of languages. To reduce the potential costs, automatic translation could be employed with human translation as a post-processing step. This dataset could serve as a natural benchmark to assess the performance of models trained with additional multilingual signal against models trained only with multilingual encoders such as Universal Sentence Encoder (Cer et al., 2018) or Multilingual BERT (Devlin et al., 2018).

Bridging the Gap Between Modular and End-to-End Modelling. The modular architecture of SDS is still a standard industrial practice for large-scale conversational AI (Ponnusamy

---

[1] Radford et al. (2019) found that extracting Reddit articles with at least three upvotes was a crucial factor for higher generative quality.

et al., 2019). However, the first edition of a task-oriented end-to-end dialogue modelling track in the DSTC8 Challenges shows the potential of more unified architectures (Kim et al., 2019). The architecture that uses the benefits of transfer learning with direct generation has been preferred by the majority of human subjects. Moreover, the challenge proved that the unification can be performed across either all modules (DST, DM and NLG) or pairs of components (DST and DM or DM and NLG). Nevertheless, the best models perform far from a satisfactory level, particularly in evaluations with real users, leaving large room for further improvements. The challenge can acquire additional levels of complexity by adding a multi-lingual layer of complexity.

Speech Conversations. This thesis primarily focused on a text as the main medium of communication. However, a vast majority of dialogue still happens through audio channels. This mode adds a layer of new challenges: 1) increased error noise due to ASR, 2) detecting the end of the speech input, and 3) backchanneling between speakers. The data collection paradigm proposed in this thesis can be adapted to voice conversations. The collection could be carried out in three ways. The first option would be to ask crowd workers to read the dialogues collected through text communication. This impairs the naturalness of the communication but poses the lowest cognitive load for the workers. The second solution would be to record one-turn replies after the crowd worker has listened to all previous turns. Finally, the workers could carry out full conversations. This allows for backchanneling and for the highest degree of freedom in interactions. However, the correctness of conversations could only be validated afterwards. The audio dataset would open the possibility of working on speech-based NLU and DM solutions that already show promising results (Lugosch et al., 2019).

I would like to conclude this thesis with a rather provocative thought. I might be that all assumptions that we take for granted when modelling task-oriented dialogues are wrong and derail us from achieving true conversational intelligence? Recent results with training an unprecedentedly large language model GPT-3 (Brown et al., 2020) makes a viable question: whether training should be rather based on demonstration and interaction. Using only demonstrations, GPT-3 achieves strong performance on many NLP tasks, including translation, question-answering, and cloze tasks, as well as shows ability to adapt to unseen domains. We can imagine that a conversational agent can be *demonstrated* a right way of interacting and primed with domain-specific key-words to steer the conversation towards user goal. The results already achieved suggest that this approach should be a viable option. However, the industry requirement of robustness and safety will pose a significant challenge that could stop from productizing models built with this paradigm.

References

Gregory D Abowd, Hung-Ming Wang, and Andrew F Monk. A formal technique for automated dialogue development. In *Proceedings of the 1st conference on Designing interactive systems: processes, practices, methods, & techniques*, pages 219–226. ACM, 1995.

Kavosh Asadi and Jason D Williams. Sample-efficient deep reinforcement learning for dialog control. *arXiv preprint arXiv:1612.06000*, 2016.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. Frames: A corpus for adding memory to goal-oriented dialogue systems. *In Proceedings of SIGDIAL*, pages 207–219, 2017. URL http://www.aclweb.org/anthology/W17-5526.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *In Proceedings of ICLR*, 2014.

Lina M Rojas Barahona, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, Stefan Ultes, Tsung-Hsien Wen, and Steve Young. Exploiting sentence and context representations in deep neural models for spoken language understanding. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 258–267, 2016.

Jon Barker, Shinji Watanabe, Emmanuel Vincent, and Jan Trmal. The fifth CHiME speech separation and recognition challenge: Dataset, task and baselines. In *Proceedings Interspeech 2018*, pages 1561–1565, 2018.

Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

Richard Bellman. The theory of dynamic programming. Technical report, DTIC Document, 1954.

Emily M Bender. The# benderrule: On naming the languages we study and why it matters. *The Gradient*, 14, 2019.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Alan W Black, Susanne Burger, Alistair Conkie, Helen Hastie, Simon Keizer, Oliver Lemon, Nicolas Merigaud, Gabriel Parent, Gabriel Schubiner, Blaise Thomson, et al. Spoken dialog challenge 2010: Comparison of live and control test results. In *Proceedings of the SIGDIAL 2011 Conference*, pages 2–7. Association for Computational Linguistics, 2011.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *International Conference on Machine Learning*, 2015.

Dan Bohus and Alexander I Rudnicky. Sorry, i didn't catch that! - an investigation of non-understanding errors and recovery strategies. In *Proceedings of SIGDIAL*, 2005.

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

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Paweł Budzianowski and Ivan Vulić. Hello, it's gpt-2–how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. *The 3rd Workshop on Neural Generation and Translation, EMNLP*, 2019.

Paweł Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Iñigo Casanueva, Lina M Rojas Barahona, and Milica Gašić. Sub-domain modelling for dialogue management with hierarchical reinforcement learning. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 86–92, 2017.

Paweł Budzianowski, Iñigo Casanueva, Bo-Hsiang Tseng, and Milica Gašić. Towards end-to-end multi-domain dialogue modelling. *Tech. Rep. CUED/F-INFENG/TR.706, University of Cambridge, Engineering Department*, 2018a. URL https://www.repository.cam.ac.uk/handle/1810/284093.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. MultiWOZ-A Large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of EMNLP*, pages 5016–5026, 2018b. URL http://aclweb.org/anthology/D18-1547.

Harry Bunt. Dimensions in dialogue act annotation. In *Proceedings of LREC*, volume 6, pages 919–924, 2006.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4517, 2019.

Iñigo Casanueva, Thomas Hain, Heidi Christensen, Ricard Marxer, and Phil Green. Knowledge transfer between speakers for personalised dialogue management. In *Proceedings of SIGDIAL*, 2015.

Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Stefan Ultes, Lina Maria Rojas-Barahona, Steve J. Young, and Milica Gašić. A benchmarking environment for reinforcement learning based task oriented dialogue management. *Deep Reinforcement Learning Symposium, Neural Information Processing Systems*, abs/1711.11023, 2017. URL http://arxiv.org/abs/1711.11023.

Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Stefan Ultes, Lina M Rojas Barahona, Bo-Hsiang Tseng, and Milica Gasic. Feudal reinforcement learning for dialogue management in large domains. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 714–719, 2018.

Asli Celikyilmaz, Silicon Valley, and Dilek Hakkani-Tur. Convolutional neural network based semantic tagging with entity embeddings. *NIPS Workshop on Machine Learning for SLU and Interaction*, 2015.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

Lu Chen, Xiang Zhou, Cheng Chang, Runzhe Yang, and Kai Yu. Agent-aware dropout dqn for safe and efficient on-line dialogue policy learning. In *Proceedings of EMNLP*, pages 2444–2454, 2017.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE, 2018.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings EMNLP*, pages 1724–1734, 2014.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Michael H Cohen, Michael Harris Cohen, James P Giangola, and Jennifer Balogh. *Voice user interface design*. Addison-Wesley Professional, 2004.

Michelle Cohn, Chun-Yen Chen, and Zhou Yu. A large-scale user study of an alexa prize chatbot: Effect of tts dynamism on perceived quality of social dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 293–306, 2019.

Heriberto Cuayáhuitl. Hierarchical reinforcement learning for spoken dialogue systems. *PhD Thesis, University of Edinburgh*, 2009.

Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira. Evaluation of a hierarchical reinforcement learning spoken dialogue system. *Computer Speech & Language*, 24(2):395–429, 2010.

Heriberto Cuayáhuitl, Simon Keizer, and Oliver Lemon. Strategic dialogue management via deep reinforcement learning. *arXiv preprint arXiv:1511.08099*, 2015.

Heriberto Cuayáhuitl, Seunghak Yu, Ashley Williamson, and Jacob Carse. Deep reinforcement learning for multi-domain dialogue systems. *NIPS Deep Reinforcement Learning Symposium*, 2016.

George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2011.

Lucie Daubigney, Milica Gašić, Senthilkumar Chandramohan, Matthieu Geist, Olivier Pietquin, and Steve Young. Uncertainty management for on-line optimisation of a pomdp-based large-scale spoken dialogue system. In *Interspeech 2011*, pages 1301–1304, 2011.

Lawrence Davis. Handbook of genetic algorithms. *CUMINCAD*, 1991.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *In Proceedings NAACL*, 2018. URL https://www.aclweb.org/anthology/N19-1423.

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. End-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777*, 2016.

Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Intell. Res.(JAIR)*, 13:227–303, 2000.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Ondrej Dušek and Filip Jurcıcek. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 45, 2016.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156, 2020.

Layla El Asri, Jing He, and Kaheer Suleman. A sequence-to-sequence model for user simulation in spoken dialogue systems. *Interspeech 2016*, pages 1151–1155, 2016.

Yaakov Engel. *Algorithms and representations for reinforcement learning*. PhD Thesis, 2005.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of SIGDIAL*, pages 37–49, 2017. URL http://aclweb.org/anthology/W17-5506.

Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. Policy networks with two-stage training for dialogue systems. *Proceedings of SIGDIAL*, pages 101–110, 2016. URL http://www.aclweb.org/anthology/W16-3613.

Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

Matthew Frampton and Oliver Lemon. Learning more effective dialogue strategies using limited dialogue move features. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 185–192. Association for Computational Linguistics, 2006.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016.

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. *Neural Information Processing Systems*, 2017.

Mark Gales, Steve Young, et al. The application of hidden markov models in speech recognition. *Foundations and Trends® in Signal Processing*, 1(3):195–304, 2008.

John S Garofolo. Timit acoustic phonetic continuous speech corpus. *Linguistic Data Consortium, 1993*, 1993.

Milica Gašić and Steve Young. Gaussian processes for pomdp-based dialogue manager optimization. *TASLP*, 22(1):28–40, 2014.

Milica Gašić, Filip Jurčíček, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. Gaussian processes for fast policy optimisation of pomdp-based dialogue managers. In *Proceedings of SIGDIAL*, pages 201–204, 2010. URL http://www.aclweb.org/anthology/W10-4334.

Milica Gašić, Filip Jurčíček, Blaise Thomson, Kai Yu, and Steve Young. On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 312–317. IEEE, 2011.

Milica Gašić, Dongho Kim, Pirros Tsiakoulis, Catherine Breslin, Matthew Henderson, Martin Szummer, Blaise Thomson, and Steve Young. Incremental on-line adaptation of pomdp-based dialogue managers to extended domains. In *Interspeech*, 2014.

Milica Gašić, Nikola Mrkšić, Pei-hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Policy committee for adaptation in multi-domain spoken dialogue systems. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 806–812. IEEE, 2015.

Milica Gašić and Steve Young. Gaussian processes for pomdp-based dialogue manager optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22 (1):28–40, 2014.

Alax Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *IEEE ASRU*, 2013.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, 2018.

Patrick Haffner, Gokhan Tur, and Jerry H Wright. Optimizing svms for complex call classification. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE, 2003.

Jack Harmer, Linus Gisslén, Jorge del Val, Henrik Holst, Joakim Bergdahl, Tom Olsson, Kristoffer Sjöö, and Magnus Nordin. Imitation learning with concurrent actions in 3d games. In *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8, 2018. URL https://ieeexplore.ieee.org/document/8490398.

Charles T Hemphill, John J Godfrey, and George R Doddington. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania*, 1990.

James Henderson, Oliver Lemon, and Kallirroi Georgila. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics*, 34(4): 487–511, 2008.

Matthew Henderson, Blaise Thomson, and Steve Young. Deep neural network approach for the dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 467–471, 2013.

Matthew Henderson, Blaise Thomson, and Jason D Williams. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, 2014a.

Matthew Henderson, Blaise Thomson, and Jason D Williams. The third dialog state tracking challenge. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 324–329. IEEE, 2014b.

Matthew Henderson, Blaise Thomson, and Steve Young. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of SIGDIAL*, 2014c.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*, 2017. URL https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/46057.pdf.

Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. Training neural response selection for task-oriented dialogue systems. In *Proceedings of ACL*, 2019. URL https://arxiv.org/abs/1906.01543.

José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang D Bui, and Richard E Turner. Black-box α-divergence minimization. *International Conference on Machine Learning*, 2016.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13. ACM, 1993.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019. URL https://arxiv.org/abs/1904.09751.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pages 1109–1117, 2016.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, 2018. URL https://www.aclweb.org/anthology/P18-1031.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Real-time inference in multi-sentence tasks with deep pretrained transformers. *arXiv preprint arXiv:1905.01969*, 2019.

Andrew J Hunt and Alan W Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376. IEEE, 1996.

Minwoo Jeong and Gary Geunbae Lee. Triangular-chain conditional random fields. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1287–1302, 2008.

Juraj Juraska, Panagiotis Karagiannis, Kevin Bowden, and Marilyn Walker. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 152–162, 2018.

Filip Jurčíček, Blaise Thomson, and Steve Young. Natural actor and belief critic: Reinforcement algorithm for learning parameters of dialogue systems modelled as pomdps. *ACM TSLP*, 7(3):6, 2011.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

John F Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41, 1984.

Kyungduk Kim, Cheongjae Lee, Sangkeun Jung, and Gary Geunbae Lee. A frame-based probabilistic framework for spoken dialog management using dialog examples. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 120–127. Association for Computational Linguistics, 2008.

Seokhwan Kim, Luis Fernando D'Haro, Rafael E Banchs, Jason D Williams, Matthew Henderson, and Koichiro Yoshino. The fifth dialog state tracking challenge. In *Spoken Language Technology Workshop (SLT), 2016 IEEE*, pages 511–517. IEEE, 2016.

Seokhwan Kim, Luis Fernando D'Haro, Rafael E Banchs, Jason D Williams, and Matthew Henderson. The fourth dialog state tracking challenge. In *Dialogues with Social Robots*, pages 435–449. Springer, 2017.

Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, et al. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*, 2019.

Young-Bum Kim, Dongchan Kim, Anjishnu Kumar, and Ruhi Sarikaya. Efficient large-scale neural domain classification with personalized attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2214–2224, 2018.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Proceedings of ICLR*, 2014.

Ilya Kulikov, Alexander H Miller, Kyunghyun Cho, and Jason Weston. Importance of a search strategy in neural dialogue modelling. *arXiv preprint arXiv:1811.00907*, 2018. URL https://arxiv.org/abs/1811.00907.

Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *Journal of machine learning research*, 4(Dec):1107–1149, 2003.

Staffan Larsson and David R Traum. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural language engineering*, 6(3-4):323–340, 2000.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

Sungjin Lee. Structured discriminative model for dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 442–451, 2013.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Xiang Li, Yaoqin Zhang, Zheng Zhang, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, et al. Convlab: Multi-domain end-to-end dialog system platform. *arXiv preprint arXiv:1904.08637*, 2019.

Fabrice Lefèvre, Milica Gašić, F Jurčíček, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu, and S Young. k-nearest neighbor monte-carlo control algorithm for pomdp-based dialogue systems. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 272–275. Association for Computational Linguistics, 2009.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of ACL*, pages 1437–1447, 2018. URL http://aclweb.org/anthology/P18-1133.

Oliver Lemon. Learning what to say and how to say it: Joint optimisation of spoken dialogue management and natural language generation. *Computer Speech & Language*, 25(2): 210–221, 2011.

Oliver Lemon and Olivier Pietquin. Machine learning for spoken dialogue systems. In *Proceedings of Interspeech*, 2007.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. Using markov decision process for learning dialogue strategies. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 1, pages 201–204. IEEE, 1998.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on speech and audio processing*, 8(1):11–23, 2000.

Jason Li, Vitaly Lavrukhin, Boris Ginsburg, Ryan Leary, Oleksii Kuchaiev, Jonathan M Cohen, Huyen Nguyen, and Ravi Teja Gadde. Jasper: An end-to-end convolutional neural acoustic model. *arXiv preprint arXiv:1904.03288*, 2019.

Jinchao Li, Baolin Peng, Sungjin Lee, Jianfeng Gao, Ryuichi Takanobu, Qi Zhu, Minlie Huang, Hannes Schulz, Atkinson Adam, and Adada Mahmoud. Results of the multi-domain task-completion dialog challenge. *TODO*, 2020.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *NAACL-HLT*, pages 110–119, San Diego, California, June 2016a. Association for Computational Linguistics.

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

Lihong Li, Jason D Williams, and Suhrid Balakrishnan. Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. In *INTERSPEECH*, 2009.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. End-to-end task-completion neural dialogue systems. In *Proceedings of IJCNLP*, pages 733–743, 2017a. URL http://aclweb.org/anthology/I17-1074.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 733–743, 2017b.

Yingzhen Li and Yarin Gal. Dropout inference in bayesian neural networks with alpha-divergences. *International Conference on Machine Learning*, 2017.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321, 1992.

Zhen-Hua Ling, Shi-Yin Kang, Heiga Zen, Andrew Senior, Mike Schuster, Xiao-Jun Qian, Helen M Meng, and Li Deng. Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends. *IEEE Signal Processing Magazine*, 32(3):35–52, 2015.

Zachary Lipton, Xiujun Li, Jianfeng Gao, Lihong Li, Faisal Ahmed, and Li Deng. Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Pierre Lison. Multi-policy dialogue management. In *Proceedings of the SIGDIAL 2011 Conference*, pages 294–300. Association for Computational Linguistics, 2011.

Bing Liu and Ian Lane. Attention-based recurrent neural network models for joint intent detection and slot filling. *Interspeech 2016*, pages 685–689, 2016.

Bing Liu, Gokhan Tür, Dilek Hakkani-Tür, Pararth Shah, and Larry Heck. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of NAACL-HLT*, pages 2060–2069, 2018. URL http://aclweb.org/anthology/N18-1187.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of EMNLP*, pages 2122–2132, 2016. URL https://aclweb.org/anthology/D16-1230.

Jaime Lorenzo-Trueba, Thomas Drugman, Javier Latorre, Thomas Merritt, Bartosz Putrycz, Roberto Barra-Chicote, Alexis Moinet, and Vatsal Aggarwal. Towards achieving robust universal neural vocoding. In *Proceedings Interspeech*, volume 2019, pages 181–185, 2019.

Ryan Lowe, Nissan Pow, Iulian V Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 285, 2015.

Bruce T Lowerre. The harpy speech recognition system. Technical report, Carnegie-Mellon University, Department of Computer Science, 1976.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. Speech model pre-training for end-to-end spoken language understanding. *Proc. Interspeech 2019*, pages 814–818, 2019.

Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

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

François Mairesse and Steve Young. Stochastic language generation in dialogue using factored language models. *Computational Linguistics*, 40(4):763–799, 2014.

James Martens. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

Baruch Mazor and BL Zeigler. The design of speech-interactive dialogs for transaction-automation systems. *Speech Communication*, 17(3-4):313–320, 1995.

Michael F McTear. Modelling spoken dialogues with state transition diagrams: experiences with the cslu toolkit. In *Fifth International Conference on Spoken Language Processing*, 1998.

Michael F McTear. Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys (CSUR)*, 34(1):90–169, 2002.

Shikib Mehri, Evgeniia Razumovskaia, Tiancheng Zhao, and Maxine Eskenazi. Pretraining methods for dialog context representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3836–3845, July 2019. doi: 10.18653/v1/P19-1373. URL https://www.aclweb.org/anthology/P19-1373.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539, 2015.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. Parlai: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84, 2017.

Teruhisa Misu, Komei Sugiura, Kiyonori Ohtake, Chiori Hori, Hideki Kashioka, Hisashi Kawai, and Satoshi Nakamura. Dialogue strategy optimization to assist user's decision for spoken consulting dialogue systems. In *2010 IEEE Spoken Language Technology Workshop*, pages 354–359. IEEE, 2010.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *Proceedings of ICML*, pages 1791–1799, 2014. URL http://jmlr.org/proceedings/papers/v32/mnih14.html.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Eric Moulines and Francis Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech communication*, 9(5-6):453–467, 1990.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1777–1788, 2017a.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association of Computational Linguistics*, 5(1):309–324, 2017b.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding, 2019.

Alice H Oh and Alexander I Rudnicky. Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*, pages 27–32. Association for Computational Linguistics, 2000.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances In Neural Information Processing Systems*, pages 4026–4034, 2016.

Tim Paek and Roberto Pieraccini. Automating spoken dialogue management design using machine learning: An industry perspective. *Speech communication*, 50(8-9):716–729, 2008.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, 2002. URL http://aclweb.org/anthology/P02-1040.

Ronald Parr and Stuart J Russell. Reinforcement learning with hierarchies of machines. In *Advances in Neural Information Processing Systems*, pages 1043–1049, 1998.

Prasanna Parthasarathi, Arvind Neelakantan, and Sharan Narang. On task-level dialogue composition of generative transformer model, 2020.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2240, 2017.

Julien Perez and Fei Liu. Dialog state tracking, a machine reading approach using memory network. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 305–314, 2017.

Denis Peskov, Nancy Clarke, Jason Krone, Brigi Fodor, Yi Zhang, Adel Youssef, and Mona Diab. Multi-domain goal-oriented dialogues (multidogo): Strategies toward curating and annotating large scale dialogue data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4518–4528, 2019.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7):1180–1190, 2008.

Roberto Pieraccini and Juan Huerta. Where do we go from here? research and commercial spoken dialog systems. In *Proceedings of SIGDIAL*, 2005. URL https://link.springer.com/chapter/10.1007/978-1-4020-6821-8_1.

Olivier Pietquin and Thierry Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(2):589–599, 2006.

Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. *arXiv preprint arXiv:1710.07654*, 2017.

Pragaash Ponnusamy, Alireza Roshan Ghias, Chenlei Guo, and Ruhi Sarikaya. Feedback-based self-learning in large-scale conversational ai agents, 2019.

Patti J Price. Evaluation of spoken language systems: The atis domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018. URL https://openai.com/blog/language-unsupervised/.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*, 2018.

Osman Ramadan, Paweł Budzianowski, and Milica Gašić. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of ACL*, pages 432–437, 2018. URL http://www.aclweb.org/anthology/P18-2069.

Carl Edward Rasmussen and Chris Williams. Gaussian processes for machine learning. 2006.

Abhinav Rastogi, Dilek Hakkani-Tur, and Larry Heck. Scalable multi-domain dialogue state tracking. *arXiv preprint arXiv:1712.10224*, 2017.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Kumar Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*, 2019.

Antoine Raux, Brian Langner, Dan Bohus, Alan W Black, and Maxine Eskenazi. Let's go public! taking a spoken dialog system to the real world. In *Ninth European Conference on Speech Communication and Technology*, 2005. URL http://www.danbohus.com/docs/letsgo.pdf.

Christian Raymond and Giuseppe Riccardi. Generative and discriminative algorithms for spoken language understanding. In *Eighth Annual Conference of the International Speech Communication Association*, 2007.

Hang Ren, Weiqun Xu, Yan Zhang, and Yonghong Yan. Dialog state tracking using conditional random fields. In *Proceedings of the SIGDIAL 2013 Conference*, pages 457–461, 2013.

Verena Rieser and Oliver Lemon. Natural language generation as planning under uncertainty for spoken dialogue systems. In *Empirical methods in natural language generation*, pages 105–120. Springer, 2009.

Verena Rieser, Ivana Kruijff-Korbayová, and Oliver Lemon. A corpus collection and annotation framework for learning multimodal clarification strategies. In *6th SIGdial Workshop on DISCOURSE and DIALOGUE*, 2005.

Verena Rieser, Oliver Lemon, and Simon Keizer. Natural language generation as incremental planning under uncertainty: Adaptive information presentation for statistical dialogue systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(5): 979–994, 2014.

Alan Ritter, Colin Cherry, and Bill Dolan. Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180, 2010. URL https://www.aclweb.org/anthology/N10-1020.

Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of SIGDIAL*, 2000.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Jost Schatzmann and Steve Young. The hidden agenda user simulation model. *TASLP*, 17(4): 733–747, 2009.

Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(02):97–126, 2006.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152. Association for Computational Linguistics, 2007.

Konrad Scheffler and Steve Young. Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In *Proceedings of the second international conference on Human Language Technology Research*, pages 12–19. Morgan Kaufmann Publishers Inc., 2002.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61: 85–117, 2015.

Nicolas Schrading, Cecilia Ovesdotter Alm, Ray Ptucha, and Christopher Homan. An analysis of domestic abuse discourse on reddit. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2577–2583, 2015.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3795–3805, 2019.

John R Searle. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge University Press, 1969.

Stephanie Seneff and Joseph Polifroni. Dialogue management in the mercury flight reservation system. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational Systems - Volume 3*, ANLP/NAACL-ConvSyst '00, pages 11–16, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/1117562.1117565. URL https://doi.org/10.3115/1117562.1117565.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808*, 2015.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of AAAI*, pages 3295–3301, 2017. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/viewPaper/14567.

Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio. Towards end-to-end spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5754–5758. IEEE, 2018.

Pararth Shah, Dilek Hakkani-Tur, Gokhan Tur, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*, 2018.

Guy Shani, Joelle Pineau, and Robert Kaplow. A survey of point-based pomdp solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51, 2013.

Heung-Yeung Shum, Xiao-dong He, and Di Li. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26, 2018.

Satinder Singh, Tommi Jaakkola, Michael L Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38 (3):287–308, 2000a.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research*, 16:105–133, 2002.

Satinder P Singh, Michael J Kearns, Diane J Litman, and Marilyn A Walker. Reinforcement learning for spoken dialogue systems. In *Advances in Neural Information Processing Systems*, pages 956–962, 2000b.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Amanda Stent, Rashmi Prassad, and Marilyn Walker. Trainable sentence planning for complex information presentations in spoken dialog systems. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 79–86, 2004.

Amanda Stent, Matthew Marge, and Mohit Singhai. Evaluating evaluation methods for generation in the presence of variation. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 341–351. Springer, 2005.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373, 2000.

Pei-Hao Su, Milica Gašić, Nikola Mrkšić, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. Continuously learning neural dialogue management. *arXiv preprint arXiv:1606.02689*, 2016.

Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gašić, and Steve J. Young. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the SIGDIAL 2017 Conference*, 2017.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proceedings of NeurIPS*, pages 3104–3112, 2014. URL https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1999. URL http://www.incompleteideas.net/book/first/the-book.html.

Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of NIPS*, volume 99, 1999.

Christopher Tegho, Paweł Budzianowski, and Milica Gašić. Benchmarking uncertainty estimates with deep reinforcement learning for dialogue policy optimisation. In *IEEE ICASSP*, 2018.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Blaise Thomson and Steve Young. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588, 2010.

David R. Traum. *Foundations of Rational Agency*, chapter Speech Acts for Dialogue Agents. Springer, 1999. URL https://www.springer.com/gb/book/9780792356011.

David R Traum and Elizabeth A Hinkelman. Conversation acts in task-oriented spoken dialogue. *Computational intelligence*, 8(3):575–599, 1992.

David R Traum and Staffan Larsson. The information state approach to dialogue management. In *Current and new directions in discourse and dialogue*, pages 325–353. Springer, 2003.

Gokhan Tur and Li Deng. Intent determination and spoken utterance classification. *Spoken language understanding: systems for extracting semantic information from speech. Wiley, Chichester*, pages 93–118, 2011.

Alan Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

Stefan Ultes, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gašić, and Steve J. Young. Pydial: A multi-domain statistical dialogue system toolkit. In *ACL Demo*. Association of Computational Linguistics, 2017.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. URL https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Oriol Vinyals and Quoc V. Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015. URL https://arxiv.org/abs/1506.05869.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700, 2015.

Marilyn A Walker. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, 12: 387–416, 2000.

Marilyn A Walker, Owen Rambow, and Monica Rogati. Spot: A trainable sentence planner. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001.

Zhuoran Wang and Oliver Lemon. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432, 2013.

Zhuoran Wang, Hongliang Chen, Guanchun Wang, Hao Tian, Hua Wu, and Haifeng Wang. Policy learning for domain selection in an extensible multi-domain spoken dialogue system. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 57–67, 2014.

Wayne Ward. Extracting information in spontaneous speech. In *Third International Conference on Spoken Language Processing*, 1994.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Jônatas Wehrmann, Rodrigo C Barros, Silvia N das Dôres, and Ricardo Cerri. Hierarchical multi-label classification with chained neural networks. In *Proceedings of the Symposium on Applied Computing*, pages 790–795, 2017. URL https://dl.acm.org/citation.cfm?id=3019664.

Wei Wei, Quoc Le, Andrew Dai, and Jia Li. AirDialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3844–3854, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1419. URL https://www.aclweb.org/anthology/D18-1419.

Gellért Weisz, Paweł Budzianowski, Pei-Hao Su, and Milica Gasic. Sample efficient deep reinforcement learning for dialogue systems with large action spaces. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 26(11):2083–2097, 2018.

Joseph Weizenbaum et al. Eliza - a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, September 2015. URL https://www.aclweb.org/anthology/papers/D/D15/D15-1199/.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. Multi-domain neural network language generation for spoken dialogue systems. *Proceedings of ACL*, 2016.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve Young. Latent intention dialogue models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3732–3741, 2017a.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of EACL*, pages 438–449, 2017b. URL http://aclweb.org/anthology/E17-1042.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. The dialog state tracking challenge. In *In Proceedings of SIGDIAL*, pages 404–413, 2013. URL http://www.aclweb.org/anthology/W13-4065.

Jason D. Williams and Steve Young. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422, 2007.

Jason D Williams and Geoffrey Zweig. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*, 2016.

Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 665–677, 2017.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. URL https://link.springer.com/article/10.1007/BF00992696.

Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*, 2019. URL https://arxiv.org/abs/1901.08149.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of ACL*, 2019a.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. Alternating recurrent dialog model with large-scale pre-trained language models. *arXiv preprint arXiv:1910.03756*, 2019b.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Xuesong Yang, Yun-Nung Chen, Dilek Hakkani-Tür, Paul Crook, Xiujun Li, Jianfeng Gao, and Li Deng. End-to-end joint learning of natural language understanding and dialogue manager. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 5690–5694. IEEE, 2017.

Takayoshi Yoshimura. Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for hmm-based text-to-speech systems. *PhD dissertation, Nagoya Institute of Technology*, 2002.

Steve Young. Talking to machines (statistically speaking). In *Seventh International Conference on Spoken Language Processing*, 2002.

Steve Young, Milica Gašić, Blaise Thomson, and Jason Williams. POMDP-based Statistical Spoken Dialogue Systems: a Review. In *In Proceedings of IEEE*, volume 99, pages 1–20, 2013a.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. POMDP-based statistical spoken dialog systems: A review. *In Proceedings of the IEEE*, 101(5):1160–1179, 2013b. URL https://ieeexplore.ieee.org/document/6407655.

Heiga Ze, Andrew Senior, and Mike Schuster. Statistical parametric speech synthesis using deep neural networks. In *2013 ieee international conference on acoustics, speech and signal processing*, pages 7962–7966. IEEE, 2013.

Heiga Zen, Keiichi Tokuda, and Alan W Black. Statistical parametric speech synthesis. *speech communication*, 51(11):1039–1064, 2009.

Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney. Improved training of end-to-end attention models for speech recognition. In *Proceedings of Interspeech*, pages 7–11, 2018.

Bo Zhang, Qingsheng Cai, Jianfeng Mao, and Baining Guo. Planning and acting under uncertainty: A new model for spoken dialogue systems. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 572–579. Morgan Kaufmann Publishers Inc., 2001.

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*, 2019.

Tiancheng Zhao and Maxine Eskenazi. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 1, 2016.

Tiancheng Zhao and Maxine Eskenazi. Zero-shot dialog generation with cross-domain latent actions. In *Proceedings of SIGDIAL*, pages 1–10, 2018. URL http://www.aclweb.org/anthology/W18-5001.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of ACL*, pages 654–664, 2017. URL http://aclweb.org/anthology/P17-1061.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Proceedings of ACL*, pages 1098–1107, 2018. URL http://www.aclweb.org/anthology/P18-1101.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1208–1218, 2019. URL https://www.aclweb.org/anthology/N19-1123.

Li Zhou and Kevin Small. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *Third Workshop in Conversational AI, NeurIPS*, 2019.

Xin-Zhong Zhu and Jian-Min Zhao. Spoken dialogue management as planning and acting under uncertainty. In *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, volume 2, pages 669–673. IEEE, 2002.

Appendix A

Comparison of Generated Dialogues

A.1 Cam676 Generation Examples

	Dialogue 1
User:	I’m looking for an indian restaurant in the south area of town.
System:	`restaurant_name` is a `value_food` restaurant in the `value_area` part of town.
User:	I don’t care. What have you found?
System:	`restaurant_name` is a `value_food` restaurant in the `value_area` part of town.
User:	Yes, please. phone number and address
System:	`restaurant_name` is located at `restaurant_address` and the phone number is `restaurant_phone`.
User:	Thank you. Bye.
System:	thank you for using the cambridge restaurant system. goodbye.
	Dialogue 2
User:	I am looking for a German restaurant in the west part of town.
System:	there are `value_count` restaurant -s in the `value_area` part of town.
User:	How about Italian?
System:	`restaurant_name` is a `value_food` restaurant in the `value_area` part of town.
User:	What is the phone number of one of them?
System:	the phone number is `restaurant_phone`.
User:	Thank you goodbye.
System:	thank you for using the cambridge restaurant system. goodbye.

Table A.1 Two dialogues from Cam676 where the system’s responses were generated by the model.

A.2 MultiWOZ Generation Examples

	Dialogue 1
User:	Hi, I am looking for a Chinese restaurant in the centre.
System:	there are `value_count` `value_food` restaurants in the `value_area`. would you like to narrow it down by price range or type of food ?
User:	Yes, I would prefer a restaurant in the moderate price range.
System:	there are `value_count` restaurants in the `value_area`. `restaurant_name` is a `value_pricerange` `value_food` restaurant in the `value_area`. would you like me to book `value_count` for you ?
User:	No, thank you. I would like the address, phone number and postcode, please.
System:	the address is `restaurant_address` and the phone number is `restaurant_phone`. is there anything else i can help you with ?
User:	Yes I'm also looking for a place to stay.
System:	i have `value_count` hotels that match your criteria. do you have a specific area you would like to stay in ?
User:	Is there anything within walking distance of the Jinling?
System:	yes , it is a guesthouse. would you like me to book a room for you ?
User:	I am looking for a hotel with free wifi available for 3 people for 3 nights starting sunday
System:	i am sorry , but there are no hotels that match your criteria. would you like to try a different area ?
User:	Actually, I'd like a moderate hotel in the centre with free wifi. I need to book it for 3 people for 3 nights starting Sunday. And a reference number please?
System:	i have booked your reservation for `value_count` people on `value_day` at `hotel_name`. your reference number is `hotel_reference`. is there anything else i can help you with today ?
User:	Yes. I'd like to book a taxi between both places. I want to leave the hotel by 20:00. Please provide me with a contact number and car type.
System:	booking completed ! booked car type `taxi_type` contact number `taxi_phone`
User:	Thank you so much, that's all I need
System:	thank you for contacting us and have a nice day.

Table A.2 First dialogue from MultiWOZ where the system's responses were generated by the model.

	Dialogue 2
User:	Hi I'm looking for lodging in Cambridge that includes free wifi and is upscale and expensive
System:	i have `value_count` hotels that match your criteria . would you like to narrow it down by area or price range ?
User:	I actually am looking for a guesthouse, not a hotel.
System:	i am sorry , there are no hotels that match your criteria . would you like to try a different area ?
User:	Is there a guesthouse that might be in the cheaper price range in the same area?
System:	there are `value_count` guesthouses that fit your needs . would you like to narrow it down by area or price range ?
User:	I need it in the east, please.
System:	i have `value_count` hotels in the `value_area` . would you like me to book `value_count` for you ?
User:	Yes. I need 3 rooms beginning on Tuesday for 2 nights.
System:	i will book it for you now .
User:	There are 3 of us.
System:	i have made your reservation for `value_count` people on `value_day` at `hotel_name` . your reference number is `hotel_reference` . is there anything else i can help you with today ?
User:	I was also wondering if you would be able to help me find a place to go to see some great architecture.
System:	there are `value_count` in the `value_area` . `attraction_name` is a nice place . would you like more information ?
User:	Can I please have the postcode for the Holy Trinity Church?
System:	sure , the postcode is `attraction_postcode` . is there anything else i can help you with today ?
User:	Okay, great. Thanks for your help!
System:	you are welcome . have a great day !

Table A.3 A second dialogue from MultiWOZ where the system's responses were generated by the model.

Appendix B

Parameters Overview of MASDA

Hyperparameter	Baselines	MASDA
GRU encoder	150	
MLP$_1$	274	-
MLP$_2$	200	-
MLP$_3$	50	-
$\mathbf{W}_1$	14	274
$\mathbf{W}_2$	274	288
$\mathbf{W}_3$	164	41
$\mathbf{W}_4$	-	274
$\mathbf{W}_5$	-	191
$\mathbf{W}_6$	-	191
GRU decoder	150	
Adam schedule - SL phase	10^{-3}	
Adam schedule - RL phase	10^{-4}	
learning rate decay	0.0	
L2 norm	10^{-5}	
gradient clipping	5.0	
dropout (training)	0.2	
SL-phase batch size	64	
RL-phase batch size	32	

Table B.1 The hyper-parameters set-up for models from Chapter 5 (Baselines) and Chapter 6 (MASDA).