HERIOT-WATT UNIVERSITY

PHD THESIS

# Data-Efficient Methods for Dialogue Systems

*Author:*
Igor SHALYMINOV

*Supervisors:*
Prof. Oliver LEMON
Dr. Arash ESHGHI

*A thesis submitted in fulfilment of the requirements for the degree of PhD*

*in the*

School of Mathematical and Computer Sciences

October 2020

The copyright in this thesis is owned by the author. Any quotation from the thesis or use of any of the information contained in it must acknowledge this thesis as the source of the quotation or information.

# Declaration of Authorship

I, Igor SHALYMINOV, declare that this thesis titled, 'Data-Efficient Methods for Dialogue Systems' and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: _____

Date: _____

*This page is intentionally left blank.*

# *Abstract*

Conversational User Interface (CUI) has become ubiquitous in everyday life, in consumer-focused products like Siri and Alexa or more business-oriented customer support automation solutions. Deep learning underlies many recent breakthroughs in dialogue systems but requires very large amounts of training data, often annotated by experts — and this dramatically increases the cost of deploying such systems in production setups and reduces their flexibility as software products. Trained with smaller data, these methods end up severely lacking robustness to various phenomena of spoken language (e.g. disfluencies), out-of-domain input, and often just have too little generalisation power to other tasks and domains.

In this thesis, we address the above issues by introducing a series of methods for bootstrapping robust dialogue systems from minimal data. Firstly, we study two orthogonal approaches to dialogue: a linguistically informed model (DyLan) and a machine learning-based one (MemN2N) — from the data efficiency perspective, i.e. their potential to generalise from minimal data and robustness to natural spontaneous input. We outline the steps to obtain data-efficient solutions with either approach and proceed with the neural models for the rest of the thesis.

We then introduce the core contributions of this thesis, two data-efficient models for dialogue response generation: the Dialogue Knowledge Transfer Network (DiKTNet) based on transferable latent dialogue representations, and the Generative-Retrieval Transformer (GRTr) combining response generation logic with a retrieval mechanism as the fallback. GRTr ranked first at the Dialog System Technology Challenge 8 Fast Domain Adaptation task.

Next, we the problem of training robust neural models from minimal data. As such, we look at robustness to disfluencies and propose a multitask LSTM-based model for domain-general disfluency detection. We then go on to explore robustness to anomalous, or out-of-domain (OOD) input. We address this problem by (1) presenting Turn Dropout, a data-augmentation technique facilitating training for anomalous input only using in-domain data, and (2) introducing VHCN and AE-HCN, autoencoder-augmented models for efficient training with turn dropout based on the Hybrid Code Networks (HCN) model family.

With all the above work addressing goal-oriented dialogue, our final contribution in this thesis focuses on social dialogue where the main objective is maintaining natural, coherent, and engaging conversation for as long as possible. We introduce a neural model for response ranking in social conversation used in Alana, the 3rd place winner in the Amazon Alexa Prize 2017 and 2018. For our model, we employ a novel technique of predicting the dialogue length as the main objective for ranking. We show that this approach matches the performance of its counterpart based on the conventional, human rating-based objective — and surpasses it given more raw dialogue transcripts, thus reducing the dependence on costly and cumbersome dialogue annotations.

# *Acknowledgements*

I am extremely grateful to my supervisors Prof Oliver Lemon and Dr Arash Eshghi: starting with the very opportunity to join Heriot-Watt University that they gave me, I have had their continued support throughout my PhD. Under their thoughtful advice, I was fortunate enough to have my efforts constantly guided while being given the freedom to pursue things that I am really passionate about.

A major part of this thesis was done during my internships at Microsoft Research, and I want to thank Dr Sungjin Lee (later with Amazon Alexa AI), Dr Hannes Schulz, Dr Alessandro Sordoni, and Adam Atkinson for the excellent mentorship I was receiving from them, as well as for the fruitful collaborations we have started together that are still ongoing.

I am very grateful to Dr Julian Hough for offering extensive consultation and constructive feedback on our disfluency detection study in Ch. 6. A special thanks goes to Dr Ondřej Dušek for his thoughtful advices on deep learning models at the very start of my PhD and for being a great mentor and collaborator in our study of response ranking for social dialogue in Ch. 8.

It was an honour to be part of the Amazon Alexa Prize team Alana and work alongside its members, fellow PhD students: Shubham Agarwal, Amanda Cercas Curry, Ioannis Papaioannou, Jose Part, Karin Sevegnani, Alessandro Suglia, Xinnuo Xu, Yanchao Yu (later an alumnus, PhD) — as well as the faculty advisors: Prof Verena Rieser and Dr Ioannis Konstas.

I was extremely fortunate to be part of Heriot-Watt University's Interaction Lab with its atmosphere of passion to change the ways people communicate with machines. I am thankful to the iLab members and alumni with whom I had a pleasure to converse about conversation: Prof Helen Hastie, Dr Emanuele Bastianelli, Dr Frank Broz, Dr Christian Dondrup, Dr Ioannis Efstathiou, Dr David Howcroft, Dr Simon Keizer, Dr Xingkun Liu, Dr Katrin Lohan, Dr José David Águas Lopes, Dr Jekaterina Novikova, Dr Andrea Vanzo, Angus Addlesee, Miltiadis Marios Katsakioris, and Weronika Sieińska.

I express deep appreciation to Heriot-Watt University for supporting me with the James Watt Scholarship which was vital for this work to happen. I acknowledge Alexa Prize 2017 and 2018 grants from Amazon which contributed to my conference travel funding. Research in Ch. 3 received financial support from the EPSRC project BABBLE (grant EP/M01553X/1).

A huge thanks goes to Louise Bingham, Tim Burns, Derek Davis, Christine McBride, Claire Porter, and Peter Scott for helping me at the organisationally challenging times of my PhD.

Finally, I am endlessly grateful to my parents Viktor and Tamara, as well as my sister Elena for their unconditional love and support throughout this journey.

# Research Thesis Submission

Please note this form should be bound into the submitted thesis.

| | | | |
|---|---|---|---|
| Name: | Igor Shalyminov | | |
| School: | MACS | | |
| Version: *(i.e. First, Resubmission, Final)* | Final | Degree Sought: | PhD, Computer Science |

## Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1. The thesis embodies the results of my own work and has been composed by myself
2. Where appropriate, I have made acknowledgement of the work of others
3. The thesis is the correct version for submission and is the same version as any electronic versions submitted*.
4. My thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5. I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6. I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

## ONLY for submissions including published works

Please note you are only required to complete the Inclusion of Published Works Form (page 2) if your thesis contains published works)

7. Where the thesis contains published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) these are accompanied by a critical review which accurately describes my contribution to the research and, for multi-author outputs, a signed declaration indicating the contribution of each author (complete)
8. Inclusion of published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) shall not constitute plagiarism.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

| | | | |
|---|---|---|---|
| Signature of Candidate: | | Date: | 01/10/2020 |

## Submission

| | |
|---|---|
| Submitted By *(name in capitals):* | IGOR SHALYMINOV |
| Signature of Individual Submitting: | |
| Date Submitted: | 01/10/2020 |

## For Completion in the Student Service Centre (SSC)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Limited Access | Requested | Yes | | No | | Approved | Yes | | No | |
| *E-thesis Submitted* ***(mandatory for final theses)*** | | | | | | | | | | |
| Received in the SSC by *(name in capitals):* | | | | | | Date: | | | | |

# Inclusion of Published Works

Please note you are only required to complete the Inclusion of Published Works Form if your thesis contains published works under Regulation 6 (9.1.2)

---

## Declaration

This thesis contains one or more multi-author published works. In accordance with Regulation 6 (9.1.2) I hereby declare that the contributions of each author to these publications is as follows:

| | |
|---|---|
| Citation details | Igor Shalyminov, Ondrej Dusek, Oliver Lemon. Neural response ranking for social conversation: A data-efficient approach. In Proceedings of the 2nd International Workshop on Search-Oriented Conversational AI, SCAI@EMNLP 2018, Brussels, Belgium, October 31, 2018, pages 1–8. |
| Author 1 | Design and implementation of the neural ranker model; experiments conducted; main author of the paper |
| Author 2 | Linear ranker model; paper contributions |
| Author 3 | Supervision; paper contributions |
| Signature: | |
| Date: | 01/10/2020 |

| | |
|---|---|
| Citation details | Igor Shalyminov, Arash Eshghi, Oliver Lemon. Challenging Neural Dialogue Models with Natural Data: Memory Networks Fail on Incremental Phenomena. In Proceedings of the 21st Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2017 - SaarDial) |
| Author 1 | Memory Network model implementation; bAbI+ augmentation technique and dataset; semantic user simulator for the BABBLE model; experiments conducted; main author of the paper |
| Author 2 | BABBLE model design and implementation; paper contributions |
| Author 3 | Supervision; paper contributions |
| Signature: | |
| Date: | 01/10/2020 |

| | |
|---|---|
| Citation details | Igor Shalyminov, Arash Eshghi,Oliver Lemon. Multi-Task Learning for Domain-General Spoken Disfluency Detection in Dialogue Systems. In Proceedings of the 22st Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2018 - AixDial). |
| Author 1 | Design and implementation of the multitask LSTM disfluency detection model; main author of the paper |
| Author 2 | Ideation of the experiments; supervision; paper contributions |
| Author 3 | Ideation of the experiments; supervision; paper contributions |
| Signature: | |
| Date: | 01/10/2020 |

| | |
|---|---|
| Citation details | Igor Shalyminov and Sungjin Lee. Improving robustness of neural dialog systems in a data-efficient way with turn dropout. The Second NeurIPS Workshop on Conversational AI: “Today’s Practice and Tomorrow’s Potential”, 2018. |
| Author 1 | Turn dropout training technique; data augmentation procedure; OOD-augmented bAbI task 6 dataset; TD-HCN, VHCN, HHCN models; experiments conducted; main author of the paper |
| Author 2 | Ideation of the experiments; supervision; paper contributions |
| Signature: | |
| Date: | 01/10/2020 |

| | |
|---|---|
| Citation details | Igor Shalyminov, Sungjin Lee, Arash Eshghi, Oliver Lemon. Data-efficient goal-oriented conversation with dialogue knowledge transfer networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 1741–1751. Association for Computational Linguistics. |
| Author 1 | Design and implementation of the DiKTNet model; experiments conducted; main author of the paper |
| Author 2 | Ideation of the experiments; contribution to the MetaLWOz dataset; supervision; paper contributions |
| Author 3 | Ideation of the experiments; supervision; paper contributions |
| Author 4 | Ideation of the experiments; supervision; paper contributions |
| Signature: | |
| Date: | 01/10/2020 |

| | |
|---|---|
| Citation details | Igor Shalyminov, Sungjin Lee, Arash Eshghi, Oliver Lemon. Few-shot dialogue generation without annotated data: A transfer learning approach. In Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue 2019, pages 32–39, Stockholm, Sweden. Association for Computational Linguistics. |
| Author 1 | Design and implementation of the FSDG model; experiments conducted; main author of the paper |
| Author 2 | Ideation of the experiments; contribution to the MetaLWOz dataset; supervision; paper contributions |
| Author 3 | Ideation of the experiments; supervision; paper contributions |
| Author 4 | Ideation of the experiments; supervision; paper contributions |
| Signature: | |
| Date: | 01/10/2020 |

| | |
|---|---|
| Citation details | Igor Shalyminov, Alessandro Sordoni, Adam Atkinson, Hannes Schulz. Fast domain adaptation for goal-oriented dialogue using a hybrid generative-retrieval transformer. In ICASSP 2020 – 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 8039–8043. |
| Author 1 | Design and implementation of the GRTr model; DSTC8 experiments conducted; main author of the paper |
| Author 2 | Contribution to the MetaLWOz dataset; supervision; paper contributions |
| Author 3 | Contribution to the MetaLWOz dataset; supervision; paper contributions |
| Author 4 | Contribution to the MetaLWOz dataset; supervision; paper contributions |
| Signature: | |
| Date: | 01/10/2020 |

| | |
|---|---|
| Citation details | Igor Shalyminov, Alessandro Sordoni, Adam Atkinson, Hannes Schulz. Hybrid generative-retrieval transformers for dialogue domain adaptation. Dialog State Tracking Challenge 8 Workshop, DSTC8AAAI 2020. |
| Author 1 | Design and implementation of the GRTr model; DSTC8 experiments conducted; main author of the paper |
| Author 2 | Contribution to the MetaLWOz dataset; supervision; paper contributions |
| Author 3 | Contribution to the MetaLWOz dataset; supervision; paper contributions |
| Author 4 | Contribution to the MetaLWOz dataset; supervision; paper contributions |
| Signature: | |
| Date: | 01/10/2020 |

| | |
|---|---|
| Citation details | Arash Eshghi, Igor Shalyminov, Oliver Lemon. Bootstrapping incremental dialogue systems from minimal data: the generalisation power of dialogue grammars. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pages 2220–2230. |
| Author 1 | Design and implementation of the BABBLE model; ideation of the experiments; main author of the paper |
| Author 2 | Implementation of the Memory Netwotk model; bAbI+ data augmentation technique and dataset; semantic user simulator for BABBLE; experiments conducted; paper contributions |
| Author 3 | Ideation of the experiments; supervision; paper contributions |
| Signature: | |
| Date: | 01/10/2020 |

| | |
|---|---|
| Citation details | Sungjin Lee, Igor Shalyminov. Contextual out-of-domain utterance handling with counterfeit data augmentation. In ICASSP 2019 - 2019 IEEE International Conference on Acoustics,<br>Speech and Signal Processing (ICASSP), pages 7205–7209. |
| Author 1 | Design and implementation of the AE-HCN(-CNN) model, experiments conducted; main author of the paper |
| Author 2 | Turn Dropout training technique; bAbI+ data augmentation technique and dataset; paper contributions |
| Signature: | |
| Date: | 01/10/2020 |

| | |
|---|---|
| Citation details | Amanda Cercas Curry, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xinnuo Xu, Ondrej Dusek, Arash Eshghi, Ioannis Konstas, Verena Rieser, Oliver Lemon. Alana v2: Entertaining and informative open-domain social dialogue using ontologies and entity linking. In 2018 Alexa Prize Proceedings. |
| Author 1 | Alana team lead; Persona bot implementation; main author of the paper |
| Author 2 | Alana v2 architecture; paper contributions |
| Author 3 | Entity Linker; Ontology bot; NLU pipeline; paper contributions |
| Author 4 | Knowledge Base-based bots; paper contributions |
| Author 5 | Neural ranker model design and implementation; data-efficient response ranking experiment; paper contributions |
| Author 6 | News and Wikipedia bots implementation; paper contributions |
| Author 7 | Faculty advisor; supervision; project ideation; paper contributions |
| Author 8 | Faculty advisor; supervision; project ideation; paper contributions |
| Author 9 | Faculty advisor; supervision; project ideation; paper contributions |
| Author 10 | Faculty advisor; supervision; project ideation; paper contributions |
| Author 11 | Faculty advisor; supervision; project ideation; paper contributions |
| Signature: | |
| Date: | 01/10/2020 |

Please included additional citations as required.

# Contents

**Declaration of Authorship** i

**Abstract** iii

**Acknowledgements** iv

**Contents** x

**List of Figures** xiv

**List of Tables** xvi

**Abbreviations** xviii

**1 Introduction** 1
- 1.1 Dialogue Systems (CUIs) . . . 2
- 1.2 The Need for Data-Efficient Dialogue Models . . . 4
- 1.3 Problems Addressed in the Thesis . . . 5
- 1.4 Contributions and Thesis Structure . . . 6

**2 Background and Motivation** 9
- 2.1 Conventional Dialogue System Architecture . . . 10
  - 2.1.1 Natural Language Understanding (NLU) . . . 11
  - 2.1.2 Dialogue State Tracking (DST) . . . 12
  - 2.1.3 Dialogue Policy . . . 12
  - 2.1.4 Natural Language Generation (NLG) . . . 15
- 2.2 Fully Data-Driven Architectures . . . 18
  - 2.2.1 Response Retrieval Models . . . 19
  - 2.2.2 Response Generation Models . . . 23
- 2.3 Key Techniques for Dialogue Response Generation Models . . . 24
  - 2.3.1 Hierarchical Response Generation Models . . . 25
  - 2.3.2 Representation Learning with Autoencoders . . . 26
  - 2.3.3 Latent Variable Models for Dialogue Response Generation . . . 29
  - 2.3.4 Attention Mechanism . . . 32
  - 2.3.5 Copy-Augmented Decoding . . . 34
  - 2.3.6 Self-Attention . . . 36
- 2.4 Transfer Learning in NLP and Dialogue Modelling . . . 39

2.4.1 Word Embedding Models . . . 39
2.4.2 Contextual Word Embeddings . . . 41
2.4.3 Transfer Learning for Dialogue . . . 42
2.4.4 Dialogue Transfer Learning Intuition: Lexical and Interactional Dialogue Similarity . . . 43
2.5 Linguistically Informed Models of Dialogue . . . 44
2.5.1 Dynamic Syntax and Type Theory with Records (DS-TTR) . . . 45
2.5.2 The BABBLE Dialogue Model . . . 46
2.5.3 Semantic User Simulation for BABBLE . . . 48
2.6 Generalisation Power and Robustness of Dialogue Models . . . 49
2.6.1 Spoken Disfluencies and Data Efficiency . . . 50
2.6.1.1 Incremental Disfluency Detection Models . . . 51
2.6.2 Out-of-Domain Robustness and Data Efficiency . . . 52
2.7 Dialogue Datasets and Data Collection . . . 53

**3 Linguistic Knowledge or Learning from Examples: A Data Efficiency Perspective 56**
3.1 Motivation . . . 56
3.2 Experimental Framework — bAbI Dialog Tasks . . . 58
3.3 The bAbI+ Dataset . . . 59
3.4 Models . . . 61
3.4.1 MemN2N . . . 61
3.4.2 DyLan: bAbI and bAbI+ Setup Details . . . 62
3.5 Experiments . . . 63
3.5.1 Experiment 1: Generalisation from Small Data . . . 63
3.5.1.1 Results: Predicting System Turns . . . 64
3.5.2 Experiment 2: Semantic Accuracy . . . 64
3.5.2.1 Results: Prediction of API Calls . . . 64
DyLan results. . . . 64
MemN2N results — small data setup. . . . 64
MemN2N results — full data setup. . . . 65
3.5.2.2 How Much Data Is Enough Data? . . . 65
3.6 Discussion . . . 66
3.6.1 MemN2N Analysis . . . 66
3.6.2 DyLan Analysis . . . 67
3.7 Conclusions . . . 68

**4 Learning Transferable Dialogue Representations 70**
4.1 Few-Shot Dialogue Generation . . . 71
4.2 The Base Model . . . 71
4.3 Dialogue Knowledge Transfer Networks . . . 72
4.3.1 Stage 1. Dialogue Representation Pre-training . . . 72
4.3.2 Stage 2. Transfer . . . 74
4.4 Baselines . . . 75
4.5 Datasets . . . 76
4.6 Experimental Setup and Evaluation . . . 77
4.7 Results and Discussion . . . 78
4.7.1 Results for the Few-Shot Setup . . . 78

4.7.2 Discussion of the Latent Representations . . . 79
4.7.3 Results with Extended Data . . . 80
4.7.4 Discussion of the Evaluation Metrics . . . 81
4.8 Conclusion . . . 82

**5 Dialogue Domain Adaptation** **83**
5.1 Fast Domain Adaptation of a Dialogue System . . . 83
5.1.1 DSTC-8, Fast Domain Adaptation Task . . . 84
5.2 Proposed Model . . . 85
5.2.1 Fine-tuning on Target Domains and Prediction . . . 86
5.2.2 Hybrid Generative-Retrieval Prediction . . . 87
5.3 Baselines and Competing Models . . . 88
5.4 Datasets . . . 88
5.5 Experimental Setup and Evaluation . . . 89
5.5.1 Human Evaluation . . . 89
5.5.2 Automatic Evaluation . . . 90
5.6 Results and Discussion . . . 91
5.6.1 Human Evaluation . . . 91
5.6.2 Automatic Evaluation . . . 92
5.6.3 Analysis of The Generated/Retrieved Responses . . . 93
5.7 Conclusion . . . 95

**6 Spoken Disfluency Detection** **96**
6.1 Motivation . . . 97
6.2 A Multitask LSTM-based Model for Spoken Disfluency Detection . . . 97
6.3 The Switchboard Dialog Acts Dataset . . . 99
6.4 Disfluency Detection Generalisation to bAbI+ . . . 100
6.5 Evaluation and Experimental Setup . . . 100
6.6 Results . . . 102
6.7 Conclusion . . . 103

**7 Improving Out-of-Domain Robustness of Dialogue Systems** **105**
7.1 The Experimental Environment for OOD Robustness Studies . . . 106
7.2 OOD-Robust Dialogue Models . . . 107
7.2.1 Hybrid Code Network Model Family . . . 107
7.2.2 AE-HCN . . . 110
7.2.3 AE-HCN-CNN . . . 111
7.3 Training with Turn Dropout . . . 111
7.3.1 TD-HCN . . . 112
7.3.2 Training the AE-HCN(-CNN) . . . 112
7.4 Experiment 1: HHCN & VHCN . . . 113
7.4.1 Results and Discussion . . . 114
7.5 Experiment 2: AE-HCN(-CNN) . . . 115
7.5.1 Experimental Setup and Evaluation . . . 115
7.6 Conclusion . . . 118

**8 Data-Efficiency in Social Dialogue** **119**
8.1 Data Efficiency in Open-Domain Dialogue . . . 120

8.1.1 The Need for Data Efficiency . . . 120
8.1.2 Users' Ratings and Explicit Feedback . . . 121
8.2 A Neural Ranker for Open-Domain Conversation . . . 123
8.2.1 Ranker Architecture . . . 124
8.2.2 Training Method . . . 125
8.3 Baselines . . . 125
8.3.1 Handcrafted Ranker . . . 125
8.3.2 Linear Ranker . . . 126
8.3.3 Dual-Encoder Ranker . . . 126
8.4 Training Data . . . 126
8.5 Evaluation and Experimental Setup . . . 127
8.5.1 Evaluation Based on Explicit User Feedback . . . 127
8.5.2 Interim Results . . . 129
8.6 Training on Larger Amounts of Data . . . 129
8.7 Discussion . . . 130
8.8 Related Work . . . 130
8.9 Conclusion . . . 131

**9 Conclusions and Future Work 132**
9.1 Directions for Future Work . . . 133

**A Dialogue Knowledge Transfer Networks — Supplementary Material 135**

**B Generative-Retrieval Transformers — Supplementary Material 139**

**C Disfluency Detection — Supplementary Material 147**

**D Data-Efficiency in Social Dialogue — Supplementary Material 148**

**Bibliography 150**

# List of Figures


1.1 Enterprise dialogue ecosystem . . . . . 2

2.1 Conventional dialogue system architecture . . . . . 10
2.2 Dialogue as a Markov Decision Process . . . . . 13
2.3 Hybrid Code Network architecture . . . . . 14
2.4 RNN-based Natural Language Generation . . . . . 17
2.5 The DSSM model architecture for web search . . . . . 19
2.6 MemN2N model architecture . . . . . 21
2.7 Sequence-to-sequence conversation model architecture . . . . . 24
2.8 HRED model architecture . . . . . 25
2.9 RNN-based autoencoder . . . . . 26
2.10 RNN-based VAE . . . . . 27
2.11 The DI-VAE/DI-VST models . . . . . 28
2.12 Latent variable Seq2Seq model . . . . . 30
2.13 The LAED architecture . . . . . 31
2.14 Visualisation of attention in machine translation . . . . . 32
2.15 Attention mechanism . . . . . 33
2.16 Pointer-Generator network architecture . . . . . 35
2.17 Pointer-Sentinel Mixture Model . . . . . 36
2.18 Transformer encoder-decoder architecture . . . . . 38
2.19 Word2Vec model architecture . . . . . 39
2.20 Example relations between Word2Vec and GloVe vectors . . . . . 40
2.21 BERT 'pretrain-finetune' architecture . . . . . 41
2.22 Interactional variations in a shopping domain . . . . . 44
2.23 Incremental parsing with DS-TTR/DyLan . . . . . 45
2.24 Processing self-corrections and clarification requests with DS-TTR/DyLan . . 46
2.25 Semantics to MDP state encoding $F$ with RT features . . . . . 48
2.26 ParlAI web interface for Wizard-of-Oz data collection . . . . . 54

3.1 bAbI Dialog Tasks structure . . . . . 58
3.2 Few-shot performance of DyLan and MemN2N . . . . . 63

4.1 DI-VAE and DI-VST (DiKTNet Stage 1) . . . . . 73
4.2 DiKTNet Stage 2 (tokens in brackets are KB data) . . . . . 75
4.3 Models' performance on the SMD dataset . . . . . 78
4.4 DiKTNet performance with extended amounts of target data used for training . 80

5.1 Example DSTC-8 support set and target dialogue in the travel domain . . . . . 84
5.2 GRTr model diagram . . . . . 85

5.3 DSTC-8 Fast Domain Adaptation — human evaluation . . . . . 90
5.4 Automatic evaluation on MetaLWOz . . . . . 91
5.5 Automatic evaluation on MultiWOZ . . . . . 92
5.6 GRTr generated/retrieved response ratios . . . . . 93

6.1 Multitask LSTM disfluency detector architecture . . . . . 98
6.2 Statistics of the SWDA corpus . . . . . 99

7.1 Hybrid Code Network model family . . . . . 107
7.2 AE-HCN model architecture . . . . . 110
7.3 Histograms of AE reconstruction scores for the bAbI6 test data . . . . . 117

8.1 Alana architecture, with an example chat . . . . . 122
8.2 Neural ranker architecture . . . . . 123
8.3 Comparison of rankers trained on extended datasets . . . . . 130

B.1 MetaLWOz pure task: a histogram of pairwise distances between generated and retrieved GRTr candidates . . . . . 146
B.2 MetaLWOz cross-task: a histogram of pairwise distances between generated and retrieved GRTr candidates . . . . . 146
B.3 MultiWOZ: a histogram of pairwise distances between generated and retrieved GRTr candidates . . . . . 146

# List of Tables


3.1 API call accuracy (%) of the MemN2N trained on the full dataset . . . . . 65
3.2 MemN2N API call accuracy (%) with extended training data . . . . . 66

4.1 Example dialogue from SMD . . . . . 71
4.2 MetaLWOz dataset statistics . . . . . 77
4.3 Stanford multi-domain dataset statistics (trainset) . . . . . 77
4.4 Selected clusters of utterances sharing the same DI-VAE codes . . . . . 79
4.5 DiKTNet's selected responses . . . . . 81

5.1 Example annotated utterances from MultiWOZ . . . . . 89
5.2 Ranking from judges' pairwise comparisons . . . . . 89
5.3 GRTr example responses . . . . . 94

6.1 Evaluation of the disfluency tagging models on SWDA . . . . . 101
6.2 Controlled generalisation evaluation on the 3 bAbI+ datasets . . . . . 101
6.3 Most common repairs in SWDA of length 1—3 . . . . . 102
6.4 SWDA repairs by POS-tag pattern . . . . . 102
6.5 SWDA repairs by interregnum . . . . . 103

7.1 Augmented dialogue example . . . . . 106
7.2 Model hyperparameters . . . . . 112
7.3 Evaluation results . . . . . 112
7.4 bAbI6, GR, and GM dataset statistics . . . . . 113
7.5 AE-HCN(-CNN) Evaluation results . . . . . 115
7.6 Performances of AE-HCN-Indep on bAbI6 Test-OOD with different thresholds 116
7.7 AE-HCN-CNN performance on bAbI6 with varying counterfeit OOD rates . . 116
7.8 AE-HCN sensitivity to the $\beta$ hyperparameter on bAbI6 OOD-augmented testset 118

8.1 Correlation study of key dialogue aspects . . . . . 121
8.2 Examples from the User feedback dataset for pairwise ranking . . . . . 128
8.3 Ranking models evaluation . . . . . 129

A.1 DiKTNet evaluation results . . . . . 135
A.2 MetaLWOz domains . . . . . 136

B.1 Automatic evaluation results on MetaLWOz pure task . . . . . 139
B.2 Automatic evaluation results on MetaLWOz cross-task . . . . . 139
B.3 MetaLWOz pure task: GRTr predictions with the closest generated/retrieved candidates . . . . . 140

B.4 MetaLWOz pure task: GRTr predictions with the most distant generated/retrieved candidates . . . 141
B.5 MetaLWOz cross-task: GRTr predictions with the closest generated/retrieved candidates . . . 142
B.6 MetaLWOz cross-task: GRTr predictions with the most distant generated/retrieved candidates . . . 143
B.7 MultiWOZ: GRTr predictions with the closest generated/retrieved candidates . 144
B.8 MultiWOZ: GRTr predictions with the most distant generated/retrieved candidates . . . 145

C.1 Multi-task LSTM training setup . . . 147
C.2 SWDA labels . . . 147

D.1 Neural rankers training setup . . . 148
D.2 VowpalWabbit ranker training setup . . . 149

# Abbreviations

| | |
|---|---|
| **AE-HCN** | Autoencoder Hybrid Code Network |
| **AMT** | Amazon Mechanical Turk |
| **API** | Application Programming Interface |
| **ASR** | Automatic Speech Recognition |
| **BERT** | Bidirectional Encoder Representations from Transformers |
| **BLEU** | Bilingual Evaluation Understudy |
| **BPR** | Batch Prior Regularisation |
| **BPTT** | Backpropagation Through Time |
| **CIDEr** | Consensus-based Image Description Evaluation |
| **CNN** | Convolutional Neural Network |
| **CRF** | Conditional Random Field |
| **CUI** | Conversational User Interface |
| **CV** | Computer Vision |
| **DiKTNet** | Dialogue Knowledge Transfer Network |
| **DI-VAE** | Discrete Information Variational Autoencoder |
| **DI-VST** | Discrete Information Variational Skip-Thought |
| **DP** | Dynamic Programming |
| **DS** | Dynamic Syntax |
| **DSSM** | Deep Semantic Similarity Model |
| **DST** | Dialogue State Tracker |
| **DSTC** | Dialog State Tracking Challenge, Dialog System Technology Challenge |
| **ELBO** | Evidence Lower Bound |
| **ELMo** | Embeddings from Language Models |
| **FT** | FastText |
| **GAN** | Generative Adversarial Network |

| | |
|---|---|
| **GloVe** | Global Vectors for Word Representation |
| **GPT** | Generative Pretrained Transformer |
| **GRTr** | Generative-Retrieval Transformer |
| **GRU** | Gated Recurrent Unit |
| **HCN** | Hybrid Code Network |
| **HMM** | Hidden Markov Model |
| **HRED** | Hierarchical Recurrent Encoder-Decoder |
| **IDF** | Inverted Document Frequency |
| **IVA** | Intelligent Virtual Assistant |
| **KB** | Knowledge Base |
| **OOD** | Out-of-Domain |
| **VHCN** | Variational Hybrid Code Network |
| **LAED** | Latent Action Encoder-Decoder |
| **LDA** | Latent Dirichlet Allocation |
| **LSTM** | Long Short-Term Memory |
| **MDP** | Markov Decision Process |
| **MemN2N** | End-to-End Memory NEtwork |
| **MetaLWOz** | Meta-Learning Wizard-of-Oz |
| **METEOR** | Metric for Evaluation of Translation with Explicit Ordering |
| **MLE** | Maximum Likelihood Estimate |
| **MLP** | Multilayer Perceptron |
| **MLM** | Masked Language Modelling |
| **MRC** | Machine Reading Comprehension |
| **MSE** | Mean Squared Error |
| **NER** | Named Entity Recognition |
| **NLG** | Natural Language Generation |
| **NLL** | Negative Log-Likelihood |
| **NLP** | Natural Language Processing |
| **NLTK** | Natural Language Toolkit |
| **NLU** | Natural Language Understanding |
| **NMT** | Neural Machine Translation |
| **NP** | Noun Phrase |
| **NSP** | Next Sentence Prediction |

| | |
|---|---|
| **POMDP** | Partially Observable Markov Decision Process |
| **POS** | Part of Speech |
| **PP** | Prepositional Phrase |
| **QA** | Question Answering |
| **ReLU** | Rectified Linear Unit |
| **RL** | Reinforcement Learning |
| **RNN** | Recurrent Neural Network |
| **RNNLM** | Recurrent Neural Network Language Model |
| **ROUGE** | Recall-Oriented Understudy for Gisting Evaluation |
| **RT** | Record Type |
| **Seq2Seq** | Sequence-to-Sequence |
| **SGD** | Stochastic Gradient Descent |
| **SMD** | Stanford Multi-Domain (dataset) |
| **SP** | SentencePiece |
| **SVM** | Support Vector Machine |
| **SWDA** | Switchboard Dialog Acts |
| **TD** | Temporal Difference |
| **TF** | Term Frequency |
| **TTR** | Type Theory With Records |
| **TTS** | Text-to-Speech |
| **ULMFiT** | Universal Language Model Fine-Tuning |
| **VAE** | Variational Autoencoder |
| **VP** | Verb Phrase |
| **WOZ** | Wizard-of-Oz |
| **ZSDG** | Zero-Shot Dialogue Generation |

*This page is intentionally left blank*

# Chapter 1

# Introduction

According to the Merriam-Webster Dictionary (2020), dialogue is "a conversation between two or more persons, also a similar exchange between a person and something else (such as a computer)". Conversation is the most versatile and efficient way for humans to communicate: a debate, a negotiation, or a friendly chit-chat — dialogue is the main means for all kinds of interaction between people. But it is not only other people whom one would want to interact with naturally: communicating with machines in a similar intuitive way has been a major goal for researchers and engineers around the world. And having past decades of Human-Computer-Interaction (HCI) research and several generations of human-computer interfaces, we are now on the brink of being able to leverage the empowering potential of computers just by talking to them. Specifically, advances in the fields of Speech Processing and Natural Language Processing (NLP) made it possible to enable interaction with an ever growing amount of devices and services using natural language — interfaces of this kind are called Conversational User Interfaces (CUIs), or dialogue systems.

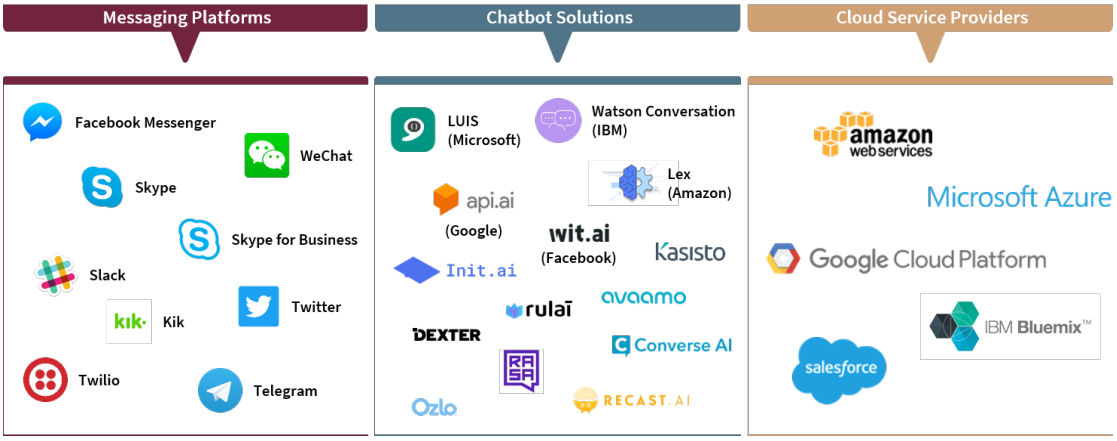
Currently, dialogue systems are already ubiquitous in everyday life:

— CUI is now enabled by most of the personal devices, e.g. Apple Siri, Amazon Alexa, Google Assistant, Microsoft Cortana, with their functionality constantly growing,

— Enterprises, especially in banking, healthcare, and retail spheres are deploying CUI solutions for automating call centres, customer support websites, and sales assistance at online marketplaces,

— Due to popular demand, a wide range of CUI building platforms are now thriving on the market, e.g. Google Dialogflow (former Api.ai), Facebook Wit.ai, Rasa, Microsoft Bot Framework, Baidu DuerOS, Amazon Lex, and Apple Siri platforms for developers. See Figure 1.1 for a visualisation of the variety of components and services in the ecosystem of dialogue solutions for enterprise.

FIGURE 1.1: Enterprise dialogue ecosystem[1]




The market for Intelligent Virtual Assistants (IVAs) and related products was valued at $2.2 billion in 2018 and expected to grow 10-fold by 2025 thus reaching $25.63 billion[2].

## 1.1 Dialogue Systems (CUIs)

Although CUI gained wide adoption in the very recent years, conversational, or dialogue systems have a long history. The first computer programs to support Natural Language interaction appeared decades ago — the first one that gained wide public recognition was ELIZA developed in 1966 by Joseph Weizenbaum at MIT (Weizenbaum, 1966). Programmed with a set of simple rules, it was tasked to maintain dialogue with the user close to what can be expected at a psychotherapy session. The system did not have any capabilities of "understanding" the user's words to any significant degree — nor was it its goal. However, in its behaviour it appeared sufficiently close to a human, and thus it was one of the first programs to attempt the Turing Test which evaluates machine's ability to simulate human behaviour so that it cannot be distinguished from a real human (Turing, 1950).

Since then, a lot of research and engineering effort has been put into creating conversational systems of ever-increasing functionality — however early systems were constrained by the limitations of the rule-based logic which was never robust and flexible enough for natural language. This has changed with the spread of machine learning techniques in the NLP field. Based on statistical analysis of real datasets rather than handcrafted rules, machine learning greatly improved dialogue systems' robustness to the aspects of spoken language as well as enabled Automatic Speech Recognition (ASR) systems to attain a practical performance level. This led to the creation of the first systems with voice interfaces — e.g. ESPIRIT SUNDIAL (Peckham, 1991)

[1] Image source: `seekpng.com`
[2] Information from GrandView Research, accessed December 2019

was one of the first spoken dialogue systems: it worked over the phone line and provided air travel information for the users. Nevertheless, its language coverage was not very wide, and apart from the state-of-the art ASR for that time, its dialogue behaviour was quite inflexible as it was based on a set of rules. However, around that time dialogue systems started gaining interest in the enterprise sphere due to their potential in optimising business processes.

Having attained sufficient performance with speech recognition (i.e. converting an utterance from audio signal into its textual form), dialogue systems research was then focused on understanding the more high-level concepts contained in the utterance, i.e. extracting *slots* (types of entities in the context of the dialogue task, e.g. 'to', 'from' for the taxi booking task) with their values, as well as detecting the overall user's *intent* in the utterance. This problem is called Natural Language Understanding (NLU) which we are going to cover in Section 2.1.1. Machine learning approaches brought significant advances to NLU, thus making conversational interfaces able to execute a wide range of voice commands in different domains. But still, the NLU logic did not cover temporal structure of conversation and the presence of the dialogue context reducing those interfaces to 1-shot interactions for the most part (some basic support of dialogue was handled by handcrafted rules).

The direction towards bringing full dialogue support to CUIs started with the introduction of Dialogue Manager (DM), a component in the dialogue system architecture (Allen et al., 2001) maintaining the conversational context in the form of a set of slots and their values provided by the user so far and taking the next dialogue action based on it (DM will be discussed in detail in Section 2.1.2). Furthermore, with the highly time-distributed nature of the conversation and extremely sparse feedback (success or failure of a dialogue can only be determined upon the end of the conversation), Supervised Learning (SL) techniques — i.e., those requiring reference output for each input to train the model — which were normally used for ASR or NLU could not be directly applied to dialogue management, and the attention of the research community shifted to Reinforcement Learning (RL) methods (Rieser and Lemon, 2011; Williams and Young, 2007) — those representing the model as and *agent* and training it from interactions with an *environment*, simulated or real-world one. However, due to the problems of scalability of early RL models and their optimisation techniques to the real-world problems, as well as the need for an excess amount of natural interactions for better training performance, Reinforcement Learning was mainly seen in academic research and proof-of-concept projects.

With the latest revolution in machine learning — the availability of massive datasets and the computational power to process them — a technique called deep learning caused a major shift in dialogue systems research, as well as the rest of the NLP field. As such, neural networks with multiple layers of neurons (thus *deep* models) trained in a unified way based on the backpropagation algorithm (Rumelhart et al., 1988) were able to learn complex patterns in raw data, their performance increasing with the amount of training examples. Specifically, one notable deep

learning-based breakthrough was the neural conversation model of Vinyals and Le (2015) — a dialogue system with a single underlying neural model trained from raw transcripts, without the use of any explicit complex features or domain knowledge. Existing approaches to dialogue system components originally powered with 'classic' SL and RL algorithms also benefited from incorporating deep learning techniques (Henderson et al., 2014c; Wen et al., 2015b; Cuayáhuitl, 2016, Li et al., 2016b). The complex patterns in the data that deep learning models were able to learn — e.g. properties of words, sentences, and paragraphs — largely replaced the need for modular architectures, by efficiently learning all the steps of conversation in a unified, *end-to-end* fashion (Serban et al., 2016).

Data has therefore become key in dialogue systems development, and the means to collect high-quality real dialogue examples came into focus (Miller et al., 2017). However, this extremely high data consumption makes deep learning models not flexible enough for the growth pace of the field — particularly limiting is the human effort in data collection and manual annotation which is both time-consuming and expensive. Moreover, these two steps have to be re-visited every time the product requirements get corrected or new functionality is desired. Therefore, it is now of key priority to develop methods for training robust and practical neural dialogue models with small amounts of data — e.g. a few example dialogues. This will keep dialogue systems being highly maintainable software products, and the corresponding development cycles rapid.

## 1.2 The Need for Data-Efficient Dialogue Models

Recent advances in dialogue modelling resulted in a massive industrial adoption of deep learning techniques. Initially originated as open-ended academic research and trained/evaluated on static large datasets, those techniques have to undergo a substantial adaptation in order to fit the industrial demands.

The first drastic difference between experimental testbeds and real-world software products is that the latter are much more dynamic and flexible. In order to stay up to date with new feature requests and tweaks for better user experience, any well-maintained product is continuously modified throughout its lifecycle. Having the core components trained directly on massive datasets, while giving them a certain level of coverage and making them generalisable to some extent, leaves the resulting models static and inflexible to the variety of target domains and usage aspects of the end products — ultimately resulting in insufficient maintainability of the latter. Apart from that, requires sensible timespans required by the large-scale training restrict the models' fitness for fast-paced development cycles. Lastly, the amount of annotations required by large-scale supervised training results in a significant financial overhead of such development strategy as well.

The above concerns can be mitigated by adopting the *data-efficient* approach to the development of neural models, i.e. enabling training from a limited amount of *seed* data. Data-efficient training assumes a series of specific problems to be solved — for dialogue systems, those are as follows. Firstly, given that deep learning methods are greatly prone to overfitting, it is of key importance for the model to attain a sufficient generalisation level outside its seed dataset. Secondly, there is a major mismatch between the specifics of written language of openly available data used for large-scale training of NLP models (e.g. internet news articles, Wikipedia documents or books) and spoken language of dialogue. The mismatch spans from differences in vocabulary and syntax to various incremental phenomena of spontaneous speech like pauses, self-corrections, restarts and other disfluencies (Hough, 2014; Healey et al., 2018). Under the end-to-end deep learning framework, the conventional solution to that would be to have all the relevant speech phenomena covered in the training data, but for low-resource training, this assumption does not hold. Finally, every dialogue system, especially the ones that provide a conversational interface to an underlying Application Programming Interface (API) or service, has the boundaries of its domain, and when working with a large user audience, it often happens that the system gets unexpected, or anomalous input outside its designated domain. Given that most such systems are trained for maximum accuracy within the domain, it is important to have a means to guarantee predictable system's behaviour outside it. And expanding the training dataset with out-of-domain data in order to train for coverage is hardly possible since all the variety of different domains and anomalous cases is very challenging to list. It is especially critical for the systems trained from minimal data, where any additional data collection comes at an especially high cost.

## 1.3 Problems Addressed in the Thesis

As outlined above, in this thesis we are going to address the following problems:

1. **Dependence on large amounts of training data with annotations.** We are going to develop methods enabling the training of goal-oriented and chat-oriented dialogue systems (as well as their key components) with minimum human effort in terms of training data collection and annotation;

2. **The lack of coverage of the diverse spoken language phenomena.** Conversational corpora collected specifically for dialogue model development do not normally represent the aspects of spontaneous spoken language, therefore the models trained on those are not ready for immediate usage in real-world settings. We will investigate ways to improve their generalisation potential to such input without the increase in the amount of the required training data;

3. **Insufficient robustness of neural dialogue models to out-of-domain input.** As is the case with training from small data, neural dialogue models are prone to overfitting to the training datasets and thus often lack robustness to anomalous out-of-domain user input which leads to unexpected dialogue behaviour and thus considerably limits the usage of such models in mission-critical production environments. We are going to improve the dialogue systems' robustness to out-of-domain input — again, without the increase in the amount of training data, i.e. only using the available in-domain data.

## 1.4 Contributions and Thesis Structure

The above problem statement defines the structure of this work: the thesis consists of a series of contributions towards practical data efficiency of machine learning-based dialogue systems. The thesis is organised as follows.

In Chapter 2 following the Introduction, we give a brief overview of dialogue system architectures — particularly, how the conventional modular architecture was transformed into the purely data-driven architecture under the influence of deep learning methods. We cover the two major types of purely data-driven architectures: response generation systems and response retrieval systems. We then outline the current state of transfer learning which is the basis for data-efficient methods in machine learning overall. After that, we give a brief background on linguistically informed models of dialogue — specifically those based on the dialogue grammars. We then discuss the problems of robustness of dialogue systems — especially those trained from minimal data — to the unseen phenomena in the input, i.e. spoken disfluencies and out-of-domain user queries. We conclude the chapter with a discussion of the dialogue datasets widely used in the field, as well as the aspects of dialogue data collection.

In Chapter 3, we conduct a motivational study of two fundamentally different approaches to dialogue: a linguistically informed model based on the semantic parser DyLan and a neural response retrieval model End-to-End Memory Network (MemN2N) — specifically, how they work in the setup of (1) natural input data containing spoken disfluencies and (2) limited training data. For (1), we introduce bAbI+, an extension to Facebook AI Research's bAbI Dialog Task 1 dataset augmented with spoken disfluencies. We outline the possible steps for obtaining the practical data-efficient solutions with either type of models. The work in this chapter was published at EMNLP 2017 (Eshghi et al., 2017, the author's contribution is the implementation of the Memory Network model, the bAbI+ data augmentation technique together with the corresponding dataset, the semantic user simulator, and the design and execution of the experiments) and SemDial 2017 (Shalyminov et al., 2017).

In Chapter 4, we focus on the problem of bootstrapping a goal-oriented dialogue system from minimal data, i.e. in a *few-shot* setup. Using the intuition that goal-oriented dialogue has a unified, domain-agnostic latent structure (e.g. sequences of dialogue acts which are sequences of characteristic words and phrases), we introduce the Dialogue Knowledge Transfer Network (DIKTNET), a model that represents that in the form of latent Variational Autoencoder codes learned from a large multi-domain source of dialogues, METALWOZ. The model outperforms the previous best approach on its target Stanford Multi-Domain (SMD) dataset. The work in this chapter was published at EMNLP 2019 (Shalyminov et al., 2019a) and SigDial 2019 (Shalyminov et al., 2019b).

In Chapter 5, we continue with the problem of training dialogue systems from minimal data as part of the Dialog System Technology Challenge (DSTC) 8 Fast Domain Adaptation task. We propose the hybrid Generative-Retrieval Transformer (GRTR). The model maintains a high diversity level by using sampling-based decoding and alternates between generation and retrieval from the support sets, part of the DSTC-8 data split. GRTR is the winning entry at the DSTC-8 Fast Domain Adaptation task as evaluated with human judges. The work in this chapter was published at the DSTC8@AAAI 2020 (Shalyminov et al., 2020) and ICASSP 2020 (Shalyminov et al., 2020).

In Chapter 6, we focus on the robustness of dialogue models to real-world variations of the user's input — specifically, spoken disfluencies (e.g. pauses, restarts, self-corrections). We propose a multitask LSTM-based model for domain-general disfluency detection improving upon the previous best model for incremental disfluency detection on open-domain Switchboard Dialog Acts (SWDA) dataset and showing superior generalisation to bAbI+. The work was presented at SemDial 2018 (Shalyminov et al., 2018b).

In Chapter 7, we go on to explore robustness to a different kind of anomalous input — *out-of-domain* (OOD) utterances. In the low-resource setting, it is cumbersome to collect real OOD data for robustness training because of the great variety of possible domains and tasks. We address this problem by (1) presenting turn dropout, a data-augmentation technique allowing training for anomalous input only using in-domain data, and (2) introducing AE-HCN, a Hybrid Code Network-based dialogue management model with an autoencoder for robust training using turn dropout. AE-HCN improves upon the standard HCN on bAbI Task 6 (DSTC-2) and the Google Multi-Domain Dialogues Dataset. The work from this chapter was presented at CAI@NIPS 2018 (Shalyminov and Lee, 2018) and ICASSP 2019 (Lee and Shalyminov, 2019, the author's contribution is the bAbI+ data augmentation procedure and the augmented bAbI Task 6 dataset, as well as the turn dropout training technique).

In Chapter 8, we turn our attention to open-domain chat-oriented dialogue and its data-efficiency aspects. Our problem here is response ranking in an open-domain ensemble-based dialogue system, which competed in the Amazon Alexa Prize. We introduce the neural response ranker

used in Alana, the 3rd place winner in the Amazon Alexa Prize 2017 and 2018. We explore two alternative supervision signals, dialogue rating and length, and show that the length-based model matches the performance of its rating-based counterpart and surpasses it given more unannotated training data, thus reducing the dependence on costly and cumbersome dialogue annotations. The work in this chapter was published at SCAI@EMNLP 2018 (Shalyminov et al., 2018a) and Amazon Alexa Prize Proceedings (Curry et al., 2018, the author's contribution is the neural ranking model and the data-efficient response ranking study).

Having introduced a series of data-efficient methods for neural dialogue systems, in Chapter 9 we conclude the thesis with an outline of future work left to be done in order to facilitate the adoption of these techniques in large-scale CUIs.

# Chapter 2

# Background and Motivation

Dialogue systems can be categorised by their purpose (goal-oriented and chat-oriented), modality of interaction (spoken, text-based, visual, multimodal) or the underlying model architecture (rule-based, data-driven), but on a high level they all have a similar structure. Over the years of research and development, this structure has undergone a series of transformations caused by the key breakthroughs affecting dialogue modelling. The most recent and influential of these transformations came with the adoption of machine learning techniques, especially deep learning. Following is a description of the dialogue system architecture and how it changed having become machine learning-centric. The structure of this chapter is as follows. Firstly, we are going to discuss the conventional dialogue system architecture (Section 2.1) and how it then transformed into the fully data-driven architectures (Section 2.2). Specifically, we will cover the two principal approaches that fully data-driven systems follow, i.e. *response retrieval* models (Section 2.2.1) and *response generation* models (Section 2.2.2), with a subsequent overview of the key techniques defining the most widely-used advances in dialogue response generation (Section 2.3). In Section 2.4, we will give a background on transfer learning, a general-purpose machine learning technique lying in the core of practical data efficiency, and the corresponding models for dialogue and NLP tasks. Having briefly covered the state of machine learning-based dialogue modelling, we then describe an alternative, linguistically informed approach to dialogue in Section 2.5. In Section 2.6, we will discuss the generalisation and robustness issues of the systems trained from minimal data. As such, we will cover robustness to disfluencies in the spoken language — and how proper processing of those phenomena is necessary for dialogue understanding. We will then discuss the problem of robustness to *out-of-domain* (OOD) input, i.e. user queries that a closed-domain dialogue system is unable to process correctly — this is especially important in the settings where no real OOD data is available during training. We will conclude the chapter with Section 2.7, a brief overview of dialogue datasets and how data collection stage is integrated into the dialogue system pipelines and frameworks.

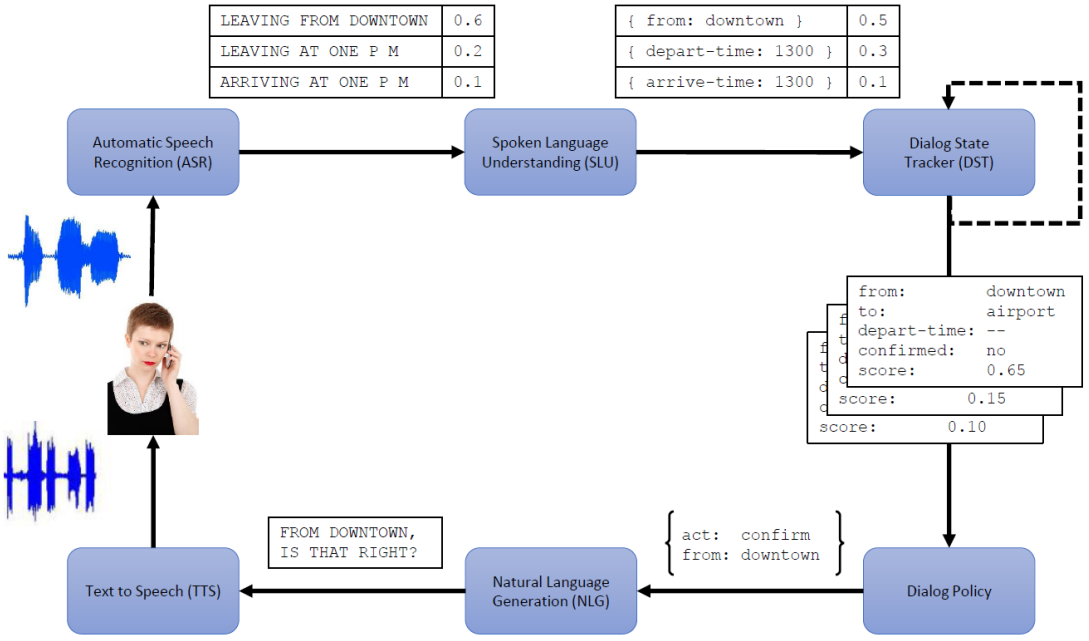



FIGURE 2.1: Conventional dialogue system architecture (Williams et al., 2016)

## 2.1 Conventional Dialogue System Architecture

Maintaining fluent and coherent dialogue assumes having solved a series of underlying problems in speech recognition/synthesis, language understanding, and action planning. The dialogue system architecture that emerged historically reflects that in its modular structure. It is normally referred to as the *conventional architecture* (Young, 2010; Williams et al., 2016) and is shown in Figure 2.1.

The conventional architecture covers the widest range of dialogue systems on a high level, including spoken and text-based ones — as such, it contains 2 components specific to voice-based interaction, namely ASR decoding audio signals from the microphone into text, and Text-to-Speech (TTS), synthesising the sound from the system's textual response. Advances in these two systems are key to the recent wide spread of personal voice assistants like Apple Siri, Amazon Alexa, and Google Assistant — and the consequent development of a large-scale market for those systems. However, audio signal processing is out of this thesis's scope. In this work, we will be focusing on the 'core' dialogue system logic that works with the user's input in the textual form — either typed in directly or already decoded by the ASR, and produces a textual response as its final output, for displaying it on the screen or feeding it into the TTS. Throughout the description of the conventional architecture, we'll be mainly talking about goal-oriented dialogue which it corresponds to most. Later in Chapter 8, we'll turn to open-domain chat-oriented dialogue as part of discussing the more recent versions of the architecture.

### 2.1.1 Natural Language Understanding (NLU)

NLU (also referred to as Spoken Language Understanding, or SLU) is the first subsystem of the conventional pipeline whose function is the extraction of relevant information from the user's input and incorporating it into the system's internal state. NLU performs this extraction on the turn level, i.e. from a single user's utterance.

Historically, dialogue follows the *frame semantics* convention for formally describing meaning (Dinarelli et al., 2009 provide a comprehensive up-to-date example). Under this notation, every situation is represented as an attribute-value frame (or form). In dialogue, a form describes a specific user's goal (also referred to as *intent*), e.g. booking a flight or searching for a restaurant by the attributes (called slots). For the flight booking task used in the diagram, the complete form contains `from`, `to`, `depart-time`, and `confirmed` slots. Therefore, goal-oriented dialogue can be represented as the form-filling process, with NLU responsible for extracting information relevant to it from a single utterance. Named Entity Recognition (NER), usually considered as a general NLP task, can be performed by the NLU as well. NER addresses the extraction of a domain-agnostic set of entities such as persons, organisations, locations or timestamps — see for example the approach of Finkel et al. (2005). Finally, as dialogue systems can handle multiple user intents with the corresponding set of tasks (e.g. 'set alarm', 'put on music', 'search web', 'chit-chat'), another type of processing performed at this stage is user intent detection. Intent detection is especially important since task-specific dialogue logic can be implemented as a completely independent subsystem — e.g. as of early 2020, Amazon Alexa contains more than 80,000 skills[1] (dialogue 'applications' within the Alexa platform) which are implemented mostly by the independent developers.

Slot value extraction is the problem that was traditionally approached using a linguistically informed method, i.e. semantic parsing relying on large-scale grammars built by linguists — e.g. the CMU Phoenix system consisted of about 13,000 rules (Tur, 2011; Ward, 1991). More recently though, the slot value extraction task shifted to the machine learning framework and was treated as a sequence labelling problem. Thus, 'classic' machine learning approaches like Hidden Markov Models (HMMs, Wang et al., 2005; Young, 2003), Conditional Random Fields (CRFs, Lafferty et al., 2001; Sha and Pereira, 2003), and Support Vector Machines (SVMs, Mairesse et al., 2009) were used in NLU predominantly. User intent detection was also initially approached using heuristic methods like keyword detection or regular expression match, but later was treated as a classification task, with the corresponding classification models like those mentioned above or efficient methods of ensembling them, e.g. *boosting* (Schapire and Singer, 2000).

[1] Information from Voicebot.ai

Among the more recent approaches to NLU, E et al. (2019) proposed a neural model for joint slot-value extraction and intent detection. The part of the model predicting slot values is a combination of a Long Short-Term Memory cell (LSTM, Hochreiter and Schmidhuber, 1997) producing the latent states and a CRF on top making the actual predictions, which itself has been widely used recently for sequence tagging tasks.

### 2.1.2 Dialogue State Tracking (DST)

The DST subsystem incorporates the information obtained from the NLU into the system's internal state (usually based on the form being filled) which it maintains throughout the dialogue. In early approaches, DST was implemented as a finite state machine maintaining the form with the new partial results coming from the NLU, and resolving any conflicts and updates caused by user's intention change or the overall ambiguity of the dialogue process. However, the actual degree of ambiguity in the real-world dialogue and fluidity of conversation with real users also made DST research shift to machine learning-based methods. As such, the annual Dialog State Tracking Challenge (DSTC) (Williams et al., 2016) was organised with the goal of advancing state-of-the-art in dialogue research. A number of approaches for DST emerged from DSTC: Recurrent Neural Networks (RNNs) for incremental word-by-word tracking (Henderson et al., 2014e; Zilka and Jurcícek, 2015), RNN-based domain-adaptive state tracking (Mrksic et al., 2015), and Convolutional Neural Networks (CNNs) for multi-language tracking (Shi et al., 2016). Since the 6th edition, with the expansion of the overall dialogue systems area, DSTC became known as Dialog System Technology Challenge, with the corresponding widening of its coverage to several tracks (e.g. 'Multi-Domain Task Completion Challenge', 'Noetic End-to-End Response Selection', and 'Visual Scene-Aware Dialogue'). As observed in DSTC-8 results, recent state-of-the-art approaches to DST tend to combine state tracking with the techniques from Machine Reading Comprehension, (MRC) (Ma et al., 2019) for better out-of-domain/zero-shot performance.

### 2.1.3 Dialogue Policy

Policy is the key component in the conventional architecture: given the accumulated system state by the DST, it makes a decision of which action to take next. Actions can be interlocutory (e.g. confirmation, information request, greeting) or functional, e.g. calls to the underlying APIs or databases. The dialogue policy is often combined with the state tracker into the Dialogue Manager (DM) component, and in the early approaches, DM was implemented as a finite state machine whose logic was to scan through the dialogue form and enquire about the next unfilled slot or, when all the necessary information is collected, issue an API call or switch to the corresponding business logic.

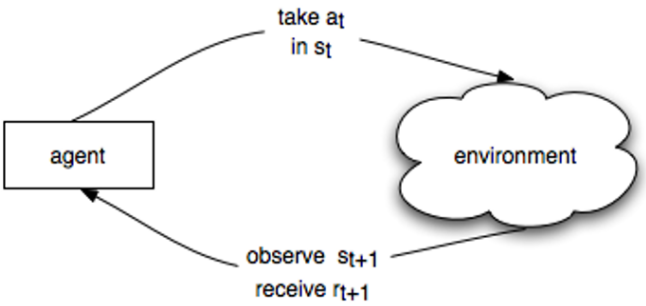



FIGURE 2.2: Dialogue as a Markov Decision Process (Rieser and Lemon, 2011)

At later stages though, it became clear that this deterministic process is not adequate to the complexity of real-world conversations, and the following two assumptions were made: (1) the information in the dialogue state is not perfectly certain nor complete, and (2) the dialogue policy should optimise an objective function defining the overall dialogue success. Thus, a policy essentially was then considered as a planning under uncertainty problem. In goal-oriented dialogue, the planning objective is normally the number of turns taken to reach the user's goal, whereas in open-domain chat-oriented dialogue it could be formulated via the user's engagement during the conversation and the overall satisfaction at the end of it (Ram et al., 2017).

In planning under uncertainty, the problems distributed in time and with a sparse delayed reward signal are modelled as Markov Decision Processes (MDPs, Rieser and Lemon, 2011; Singh et al., 2002) or Partially Observable MDPs (POMDPs, Young et al., 2013; Williams and Young, 2007), with RL as the optimisation framework. Naturally, an MDP is defined as a tuple $< S, A, \mathcal{T}, \mathcal{R} >$ — the state space, the action space, the transition function between states, and the reward function. The dialogue system thus becomes an *agent* communicating with the *environment* (user) by making observations, performing *actions* and receiving the corresponding *rewards*. The form maintained by the DST becomes the agent's internal state, and the agent's goal is to go from the initial state (empty form) to the final state (form filled out, API call issued, information presented to the user) via an optimal (or sufficiently close to such) series of actions defined by the learned policy — the procedure is visualised in Figure 2.2.

In turn, the RL framework accounts for the time-distributed nature of the dialogue as well as sparse, delayed reward. Specifically, in RL we operate on the *value* of each state — that is, the expected cumulative (final) reward that the agent will get starting at a certain state $s$ at the timestep $t$ and following the optimal policy $\pi$:

$$V^{\pi}(s) = \mathbb{E}_{\pi}(\mathcal{R} \mid s_t = s) \tag{2.1}$$

The values for each state are obtained via solving Bellman equations (Bellman and Kalaba,

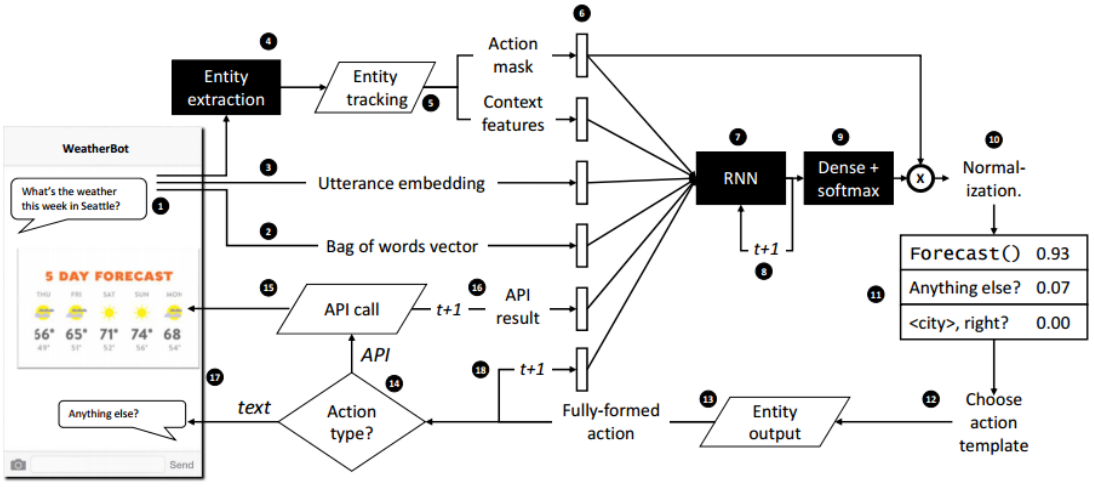



FIGURE 2.3: Hybrid Code Network architecture (Williams et al., 2017). The black parts are trainable components (with entity extraction being an independent NLU engine); trapezoids denote the parts of in-domain logic implemented by the engineers.

1957) in a Dynamic Programming (DP) way, or by using simulation-based methods, e.g. Temporal Difference (TD). TD-based methods such as Q-learning or SARSA are more widely used as they do not require knowing the dynamics of the model, i.e. all the state transition probabilities.

Successful application of TD learning assumes having the source of interactions for the agent, e.g. a simulated environment or an embodiment for real-world interaction. In some cases, especially videogames (Mnih et al., 2013; Berner et al., 2019), it is feasible to obtain lots of real-world interactions to run a sufficient amount of training episodes. In dialogue however, such data may only come from the interactions with real users which requires an existing deployment of a prototype system. Therefore, a lot of research effort was made to develop efficient user simulators for 'bootstrapping' the initial policy (Rieser and Lemon, 2011; Shi et al., 2019a), but in practice, training an efficient simulator is as hard as training the final dialogue system.

More recent approaches to Dialogue Management use supervised learning techniques such as Recurrent Neural Network-based (RNN) Hybrid Code Networks, or HCNs (Williams et al., 2017) and End-to-End Memory Networks, or MEMN2Ns (Sukhbaatar et al., 2015; Bordes et al., 2017): trained with long enough context, these models approximate long-term optimal behaviour well enough for practical usability in relatively short conversations.

The Hybrid Code Network is shown in Figure 2.3 — it is a recent example of a neural network-based dialogue management model. The overall system architecture follows the modular approach as it has separate components for entity extraction (for which it used the LUIS system by Williams et al., 2015) and template-based Natural Language Generation (we'll cover this subsystem in the next section). HCN only focuses on state tracking and action selection, and it

is aimed at training from minimal amounts of data for the direct use in products. To ensure the stability in real-world interactions in the setting of minimal training data, HCN introduces the concept of *action masks* which may be considered as expert rules incorporated into a machine learning-based model. Action masks, the binary masks applied at the DM's output, prohibit the system from issuing infeasible actions at the critical points in the dialogue, e.g. issuing an API call of bank transfer before confirming the recipient account with the user. These actions have to be hand-crafted by the domain experts and incorporated into the training pipeline. Overall, this hybrid architecture allows to alternate the emphasis between the tight, handcrafted control over the system's behaviour via action masks and the flexibility of learning from examples. The authors presented a 2-stage approach to train an HCN: at the first stage, the model learns to mimic the training dialogue examples in a supervised way, and at the second stage, the system can be further fine-tuned autonomously, from the interactions with the users in a reinforcement learning fashion. For that, the authors used a *policy gradient* approach (Williams, 1992) with the following gradient update:

$$w \leftarrow w + \alpha \left( \sum_t \nabla_w \log \pi(a_t \mid h_t; w) \right) (G - b) \tag{2.2}$$

In the above formula, the gradient is applied to the policy LSTM $\pi$ producing a distribution over actions $a$ at the timestep $t$ given the dialogue history $h_t$. The error is the difference between $G$, the return of the dialogue (the expected discounted sum of rewards), and $b$, baseline average return set heuristically. This switch from supervised to reinforcement learning (referred to as *continuous learning*) of the same exact model proved to be the most widely-used way to incorporate reinforcement learning into the dialogue system training setups (Su et al., 2017; Su et al., 2016).

HCN lies in the core of Microsoft Conversation Learner (Shukla et al., 2020), a tool for rapid prototyping of goal-oriented dialogue systems from example conversations which assumes data-efficient training. We are going to re-visit the model in Chapter 7 to explore the problem of handling out-of-domain user input unseen during training in the setting where no training examples like that are available.

### 2.1.4 Natural Language Generation (NLG)

The last processing stage in the conventional, modular dialogue system architecture is the NLG. At this stage, with the system's action chosen by the policy, the surface form of the corresponding utterance is generated. As shown in Figure 2.1, the information coming from the policy is the dialogue act identifier itself as well as some of its attributes. The reason this information is passed into NLG is that traditionally, response generation is considered a 2-stage process:

firstly, a *delexicalised* template of the final utterance is generated or selected (i.e., with all the case-specific slot values replaced by filler tokens like <slot_name>). After that, the template gets lexicalised back into its final surface form using the extra information from the policy. One of the most widely used notations for such frames is CUED Standard Dialogue Acts (Young, 2009).

In early, template-based approaches to NLG, templates were stored explicitly, and the system would just pick a random one corresponding to a certain dialogue act (e.g. Rudnicky et al., 1999). Later on, NLG was considered as an optimisation problem (similarly to the components earlier in the pipeline discussed above), inspired by the advances in artificial intelligence and planning under uncertainty. NLG's optimisation objectives were those of the information presentation problem, i.e. maintaining the users' focus, speeding up information exchange, and improving the overall task success rate. For example, an adaptive NLG component can learn how to present a database lookup result with 1, 5, or 50 results — that is, whether to go one by one or limit the output to the top-3, or announce the number of results and just display the top result. As every dialogue assumes making a series of such information presentation decisions, with the feedback coming at the very end, it is intuitive to approach this problem within the RL framework. There has been a number of works proposing RL methods for NLG, e.g. Dethlefs and Cuayáhuitl (2010), Rieser and Lemon (2009). They mainly focused on learning the 1st generation stage, i.e. response planning, while still heavily relying on rule-based surface realisation, thus being limited in the overall output's flexibility. Dusek and Jurcicek (2015) used a hybrid approach for planning: the $A^*$ algorithm for the syntactic dependency tree construction, with perceptron-based pruning (their surface realisation stage for constructing sentence plans in the shape of syntactic dependency tree was also rule-based).

The most important transformation to statistical NLG happened with the coming of deep learning-based methods: recurrent neural networks (RNN, LSTM) allowed to streamline the process by directly generating sentences word-by-word in a language model way, that is generating the next word given the context, e.g. RNNLM (Mikolov et al., 2010) — with the dialogue act information stored in the latent network state. A basic RNN-based NLG model is shown in Figure 2.4. The model generates output words by maintaining its hidden state $h$ and updating it with every input token as follows (the equations below correspond to the particular model known as Elman RNN, Elman, 1990):

$$h_t = \sigma_h \left( W_h x_t + U_h h_{t-1} + b_h \right) \tag{2.3}$$

$$y_t = \sigma_y \left( W_y h_t + b_y \right) \tag{2.4}$$

where $x_t$ is the input at time step $t$ (in our case, an encoded token), $y_t$ is the model's corresponding output, $h_{t-1}$ is the model's previous state, $W_h, W_y, U_h, b_h, b_y$ are the model's trainable

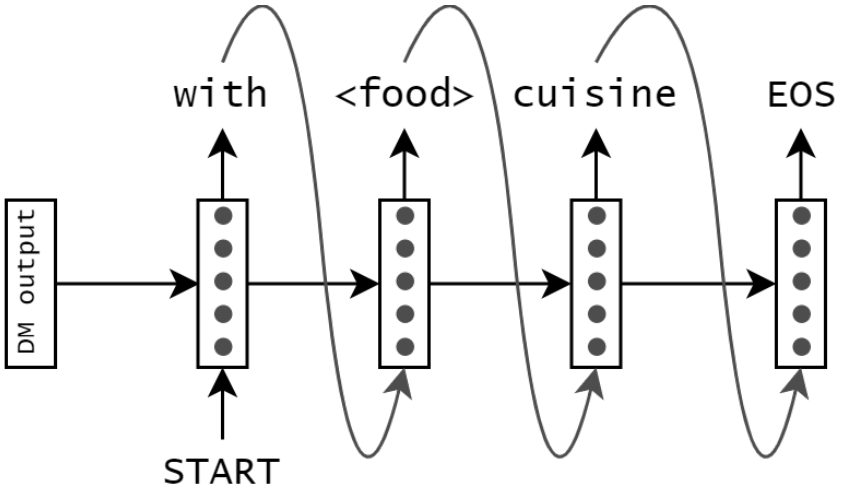



FIGURE 2.4: RNN-based Natural Language Generation

weights, and $\sigma_h, \sigma_y$ are activation functions. Usually, sigmoid activation is used for the hidden state and softmax is used for the output. The initial state $h_0$ of the model in the picture is the DM output (normally a dialogue act). This information can also be passed into the network at every step, e.g. as in Wen et al. (2015a). Note that in the picture, the model's input $x_t$ is its previous output $y_{t-1}$. This is the setup such models operate in at inference time, however at training time, $x_t$ may as well be the ground truth tokens (an approach referred to as teacher forcing).

In practice, basic RNN networks were quickly replaced by LSTMs (Hochreiter and Schmidhuber, 1997) which improve upon RNNs in training stability via using trainable *input, output, and forget* gates directly controlling what information to store in the model's state and what to explicitly forget. In addition, the approach of Wen et al. (2015b) combined two RNN cells, one being a regular LSTM for input tokens, and the other being a special 'lightweight' Dialogue Act cell.

The Sequence-to-Sequence (SEQ2SEQ) neural architecture (Vinyals and Le, 2015; Sutskever et al., 2014; Cho et al., 2014b, to be discussed in detail later in Section 2.2.2) also allowed to both encode the dialogue act into a latent representation and then decode the output in a unified token-by-token process. Dusek and Jurcícek (2016) used this approach to generate both syntax trees for the further surface realisation and the fully realised output sentences.

The encoder-decoder SEQ2SEQ architecture represented the fully data-driven approach to NLG which assumes having a single streamlined training/prediction procedure with no intermediate stages and supervision. This approach was adopted with other dialogue system components, e.g. dialogue managers went through a similar transformation from having 2 separate subsystems, DST and Policy, to a unified architecture like the HCN model discussed above. Eventually, the entire dialogue system pipeline became fully data-driven. In the next section, we are going to

discuss the key types of dialogue system architectures that emerged as part of the fully data-driven (Gao et al., 2019), or corpus-based (Jurafsky and Martin, 2019) approach to dialogue modelling.

## 2.2 Fully Data-Driven Architectures

'Classic' machine learning methods transformed the dialogue systems field so that the most efficient approaches shifted from analytical methods like expert rules, grammars, and ontologies towards data-driven techniques, e.g. CRFs for NLU or reinforcement learning over (PO)MDPs for dialogue management. This resulted in more flexible and adaptive systems, and also in a shift of focus in their development towards, firstly, collecting data (corpora of hundreds of dialogues were normally used at that point) and secondly, tasking experts with feature engineering for the machine learning models instead of directly writing the rules for the system. This reliance on feature engineering instead of task-specific expert knowledge was the main factor contributing to the overall success and efficiency of the machine learning approach.

With the arrival of large, internet-scale datasets and computational power to train machine learning models large enough to make use of this data, the next generation of machine learning — deep learning — had started. Characterised by using multi-layer (or deep) model architectures and the unified technique for training them (predominantly variants of Stochastic Gradient Descent, SGD , Bottou, 2010), deep learning approaches transformed the machine learning framework in the following fundamental way. With enough training data, deep neural networks were able to approximate non-linear relations between the input variables, thus learning features — often hierarchical — without the need for manual engineering (LeCun et al., 1998). This opened the possibility to train models for the target tasks directly from raw data, with the underlying latent features learned automatically and more efficiently. One of the first and most notable examples of training a neural network from raw data was the approach of Mnih et al. (2013) featuring an agent for playing Atari games trained with Deep RL directly from the pixels of the game screen. This advance also had a massive impact on NLP: the benefits of training from more data with less annotations were experienced in machine translation, question answering, document summarisation. Dialogue systems research was one of those areas transformed under the influence of deep learning methods. As such, the models became less modular, with the main focus on collecting a large amount of conversations and training the the entire model (or its core part) on it. We are going to describe several kinds of such models below.

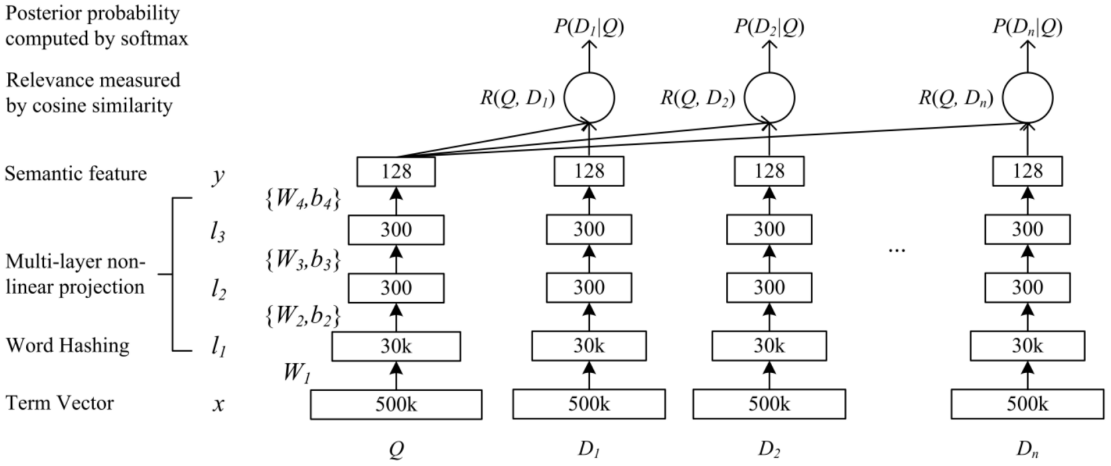



FIGURE 2.5: The DSSM model architecture for web search (Huang et al., 2013). Semantic representation of indexed documents $D_i$ as well as the user's query $Q$ is obtained with multiple trainable non-linear projection layers. The relevance scores are obtained via cosine similarity of query/document encoding pairs, then turned into a probability distribution over documents.

### 2.2.1 Response Retrieval Models

Producing conversational utterances may be considered a response selection task, when a dialogue system works similarly to a search engine: indeed, searching in a collection of documents for those relevant to a user's query is analogous to searching for utterances given user's input (possibly together with the dialogue context). Dialogue models working in this way are called *response retrieval* models. They have a search database (or *index*) of responses or full conversations, and given a new dialogue context at the input, they retrieve from the index the best candidate given a certain optimality criterion. This criterion can be the similarity between the context and the response, e.g. TF-IDF or Okapi BM25 (Manning et al., 2008) — or some more advanced learned objective functions.

Retrieval models are widely used in *chat-oriented* dialogue where the objective is to maintain the conversation and keep the user engaged and entertained. Naturally, chat-oriented systems can benefit from conversational data openly available on Internet, e.g. discussion forums (Baumgartner et al., 2020), movie subtitles (Lison and Tiedemann, 2016) or post threads on social networks (Sordoni et al., 2015). One example of such architecture is the Deep Semantic Similarity Model (DSSM) (Huang et al., 2013) which was originally developed for document re-ranking in web search. The DSSM architecture is visualised in Figure 2.5.

DSSM is a deep feed-forward neural architecture: the input, a bag-of-words term vector $x$ (corresponding to a search query $Q$ or to a document $D_i$, $i = 1, \ldots, n$ in the search database) is fed through a series of non-linear layers $l_1, \ldots, l_n$ with the trainable weight & bias parameters $W_1, \ldots, W_n$ and $b_1, \ldots, b_n$, respectively — ultimately resulting in a deep semantic representation $y$ of the input. Formally, this pipeline is of the following form:

$$
\begin{aligned}
l_1 &= W_1 x \\
l_i &= f\left(W_N l_{i-1} + b_i\right),\ i = 2, ..., N-1 \\
y &= f\left(W_N l_{N-1} + b_N\right)
\end{aligned}
\tag{2.5}
$$

where $f$ is the tanh activation function:

$$
\tanh(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)} \tag{2.6}
$$

The semantic relevance of a document $D$ given the query $Q$ is then calculated as a cosine distance:

$$
\begin{aligned}
R(Q, D) &= \cos(y_Q, y_D) \\
&= \frac{{y_Q}^{\mathsf{T}} y_D}{\left\|y_Q\right\| \left\|y_D\right\|}
\end{aligned}
\tag{2.7}
$$

The technique for calculating document relevance given a query in deep semantic representation was translated into the conversational response selection framework: as such, it was used in Microsoft Research's XiaoIce bot (Zhou et al., 2020). DSSM was used there as the response re-ranker — that is, having retrieved the initial set of response candidates over a large collection of conversations using simple and 'fast' relevance metrics like TF-IDF or Okapi BM25, those candidates were then re-ranked using the more fine-grained (and computationally 'heavy') model. The same 2-stage retrieval process is used in search engines, with the re-ranker constantly pushed to work faster in order to handle a greater number of documents with real-time performance, eventually taking place of the main ranking stage (Zamani et al., 2018).

Later on, with the development of neural architectures more suitable for textual data (predominantly LSTMs), new DSSM-based response selection models emerged. As such, QA-LSTM (Tan et al., 2015) initially introduced for the Question Answering (QA) task was later used as the response ranker in the personal chatbot Replika (Fedorenko et al., 2018) which gained massive adoption on the mobile application market.

Another widely-used family of neural retrieval architectures is Memory Networks (MemNNs, Weston et al., 2015) which are based on the notion of explicit memory, with the network operating a set of 'cells' storing observations e.g. supporting facts for QA or context utterances for dialogue. Part of the architecture was also a differentiably trainable controller for reading/writing the memory. The end-to-end variant of memory networks, MemN2N (Sukhbaatar et al., 2015) was the successor of MemNNs for the QA task — but later on, it was adapted to dialogue response selection. Therefore, we will explain the intuition behind the model for both QA and dialogue.

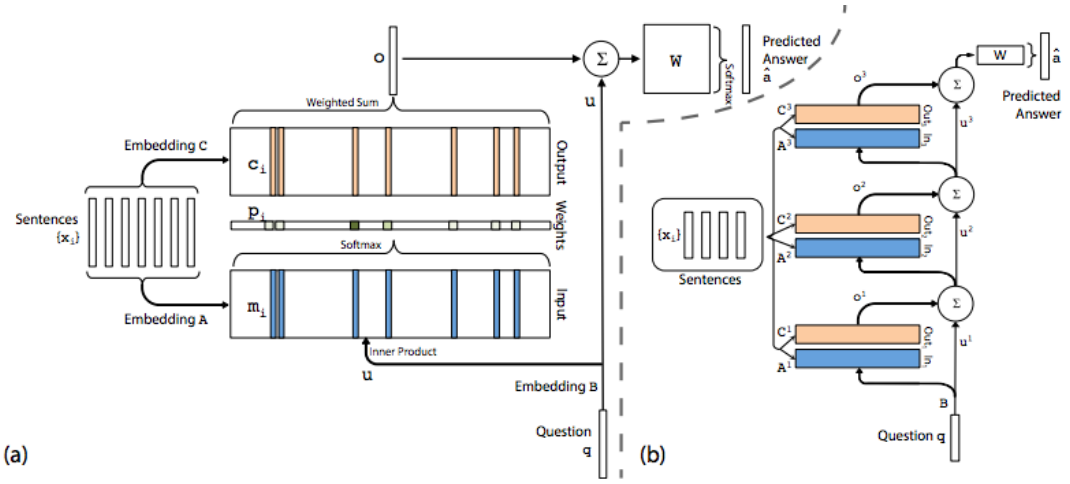



FIGURE 2.6: MEMN2N model — single-hop (a) and multi-hop (b) architectures (Sukhbaatar et al., 2015)

The main MEMN2N architecture is shown in Figure 2.6 (a) — its main components are embedding matrices $A$, $B$, and $C$ providing the differentiable representations for memories and the user's query and the output projection matrix $W$. Specifically, input sentences (context in case of dialogue or facts in case of QA) are represented in the form of memories using the embedding matrix $A$. The same is done with $q$ — user's utterance in case of dialogue or the question in the QA setting — using the matrix $B$. Then, similarly to DSSM and QA-LSTM, the key operation is calculating the similarity between the user's query and each of the memories:

$$p_i = \text{softmax}\left(u^\top m_i\right) \tag{2.8}$$

where $\text{softmax}(x) = \frac{1}{1+\exp(-x)}$, $u$ is the embedded user's query, and $m_i$ are system's memory cells.

These similarity scores implement what is originally referred to as *reading from memory with attention*. That is, the contents of the memory cells contribute to the final answer proportionally to their attention weights $p_i$ (we'll also discuss other variants of the attention mechanism in detail in Section 2.3). Next, MEMN2N takes a weighted sum of the memories using the output embedding $C$ and weights $p_i$:

$$\begin{aligned} c_i &= x_i{}^\top C \\ o &= \textstyle\sum_i p_i c_i \end{aligned} \tag{2.9}$$

The final operation in MEMN2N is similar to DSSM or QA-LSTM: a relevance metric is calculated over the user's query $u$ together with the combined system's memory state $o$ and all the actions available for the system $y_i$ (the set $Y$ includes all the possible responses and API calls):

$$\hat{a} = \text{softmax}\left((o + u)^{\mathsf{T}} \cdot W(y)\right) \tag{2.10}$$

Unlike DSSM and QA-LSTM, cosine similarity is not used as the relevance metric in this case: output matrices $W$ and $C$ serve this purpose instead (as well as $A$ and $B$, as in end-to-end training, all the model components contribute to the final task). The crucial difference between QA and dialogue setups here is that in the QA task, answers correspond to single vocabulary words, so the projection $W$ produces a distribution over the vocabulary ids. In dialogue however, response candidates are multi-word utterances, with different sets for training and testing. Therefore, they cannot be predicted as ids, and are instead embedded using the matrix $W$ as shown in Eq. 2.10.

Described above is a 'single-hop' MemN2N; the architecture however can be extended to multiple 'hops'. Such deep MemN2N is visualised in Figure 2.6 (b), with separate $A_i$ and $B_i$ matrices for every hop. A multi-hop model is basically a stack of base MemN2Ns connected in the following way: the combined system/user state $o_i + u$ — before the output projection — is passed onto the $i + 1$'th hop as the input, and the projection $W$ is only applied at the final hop for producing the final answer.

The above model was used by Bordes et al. (2017) to train a goal-oriented dialogue response retrieval system only using raw utterances. The model was used within a synthesised dataset bAbI Dialog Tasks, an experimental testbed designed to shed light on the complex problem of goal-oriented dialogue management by decomposing it into several tasks of increasing complexity. By showing quite impressive behaviour of MemN2N on synthesised still challenging data (the action set of bAbI Dialog Tasks exceeded 4,000), Bordes et al. demonstrated that goal-oriented dialogue which has traditionally been considered complicated, mission-critical task and relied on an extensive pipeline of components like language understanding, state tracking, response planning (all described earlier in this chapter), and integration with domain-specific APIs can potentially be solved with a single unified model with raw example dialogues at the input. We are going to look into MemN2N's performance within the testbed of bAbI Dialog Tasks in Chapter 3 to see how specific surface variations of the user's input unseen at training time can affect the model's performance.

There emerged more Memory Network-based models later on, including knowledge-based (Ganhotra and Polymenakos, 2018), personalised (Luo et al., 2019), Key-Value MemNNs (Miller et al., 2016), as well as those pipelined in a 'retrieve-and-refine' architecture (Weston et al., 2018).

Although response retrieval models gained wide adoption in industrial applications and products because their output is fluent and more predictable — dependent on the quality of the corpus of the candidate responses — their flexibility is always bounded by the corpus at the same time. This limitation is addressed in a parallel line of dialogue systems research, focused on *response*

*generation*. Being significantly harder to train and providing less certainty about the output fluency, models of this type are theoretically capable of generating novel utterances word-by-word and are most extensively studied in academia. We are going to discuss them in detail below.

### 2.2.2 Response Generation Models

Response generation approach to conversation modelling stemmed from an advancement in Neural Machine Translation (NMT), namely the SEQ2SEQ model (Sutskever et al., 2014). This LSTM-based model was one of the first end-to-end approaches to machine translation, working with raw text tokens as opposed to previous phrase-based (e.g. Koehn et al., 2003) and syntax-based (e.g. Yamada and Knight, 2001) models. Apart from machine translation, SEQ2SEQ became a general-purpose model for various NLP tasks due to its versatility: it was able to take a sequence of raw tokens at the input, encode it in a 'thought vector' and produce an output sequence of a length independent from the input (referred to as a *many-to-many* model). Arguably the most ambitious application of SEQ2SEQ was in training an open-domain conversation model, the problem that has never been approached before with a single, purely data-driven machine learning model (Vinyals and Le, 2015).

As opposed to goal-oriented dialogue with the specific objective to reach the user's goal with the minimum amount of conversational exchanges, open-domain conversation does not have a certain notion of the goal. Therefore, the initial objective that the dialogue SEQ2SEQ was trained with is to essentially mimic the responses seen in a large training set of conversations, e.g. a corpus of movie subtitles (the OpenSubtitles corpus of Tiedemann, 2009 was used in the original work). More formally, this objective is expressed as the Maximum Likelihood Estimate (MLE), of the output tokens $y_1, ..., y_{T'}$ given the input tokens $x_1, ..., x_T$:

$$p(y_1, ..., y_{T'} \mid x_1, ..., x_T) = \prod_{t=1}^{T'} p(y_t \mid v, y_1, ..., y_{t-1}) \tag{2.11}$$

The formula above reflects the *encoder-decoder* nature of SEQ2SEQ: the encoder component (an RNN-based model in the original work) processes the input sequence $x_1, ..., x_T$ word by word, eventually producing a latent representation of the input $v$. The decoder component (also RNN-based) produces the output sequence $y_1, ..., y_{T'}$ word by word, at each step based on the internal RNN state and the last generated token. Starting with $v$ as its initial state, the decoder updates and maintains it throughout the generation process. Crucially, the decoder's output sequence is not limited to any fixed length — instead, the decoder is directly trained to produce a special *end-of-sequence* <EOS> symbol at which the output effectively ends for all the downstream processing. As we noted above, in the original paper, both encoder and decoder were recurrent

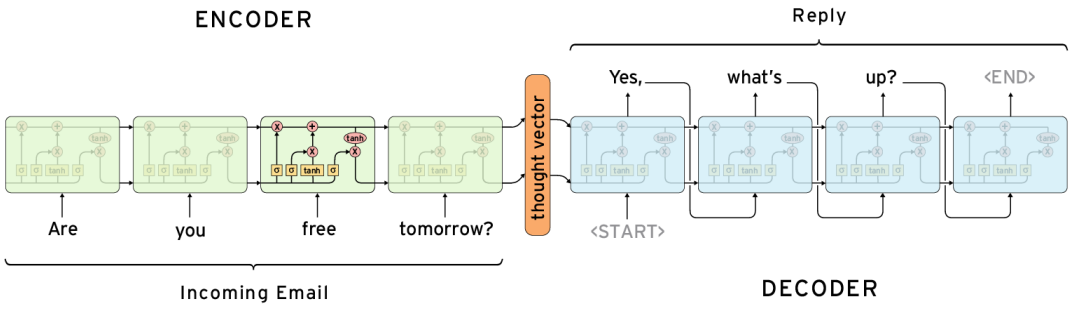



FIGURE 2.7: Sequence-to-sequence conversation model architecture[2]

networks — specifically, LSTMs, but in general, the requirements to SEQ2SEQ components are as follows:

1. the encoder is any 'many-to-one' model, i.e. able to encode an input sequence into a vector (in later versions of the model, a 'many-to-many' model outputting all the intermediate word-by-word encodings is required),
2. the decoder is any 'one-to-many' model able to generate an output sequence out of a single vector.

A SEQ2SEQ conversation model based on LSTMs is visualised in Figure 2.7. As mentioned above, the model is trained to 'mimic' responses from the training dataset by learning from randomly sampled context-response pairs over large conversational corpora. Initially, datasets of movie subtitles were used for that, e.g. Cornell Movie Dialogs dataset (Danescu-Niculescu-Mizil and Lee, 2011), OpenSubtitles (Lison and Tiedemann, 2016) — as well as threads of posts on message boards, e.g. Reddit conversations (Baumgartner et al., 2020) and Twitter comments (Sordoni et al., 2015).

## 2.3 Key Techniques for Dialogue Response Generation Models

The SEQ2SEQ architecture was one of the most transformative advances in dialogue modelling, and it was followed by a multitude of approaches building on top of it and improving it in various ways. Specifically in case of dialogue, there were a number of aspects in which it could be improved as the model did not account for the turn-taking nature of the dialogue and the overall objective of a successful conversation still remained to be formulated better. Therefore, in this section we will be focusing on the key improvements that made SEQ2SEQ model represent dialogue more adequately.

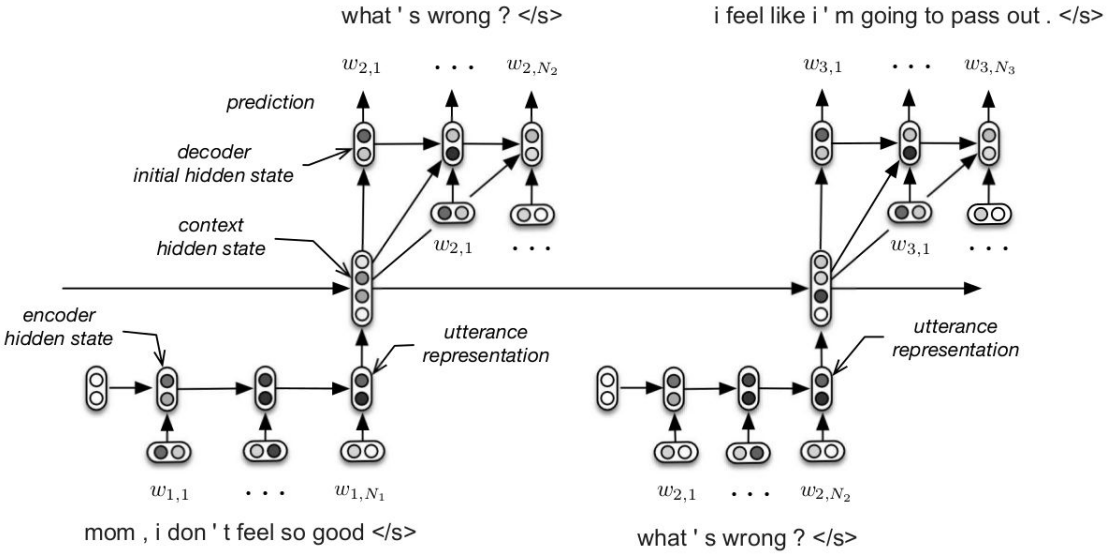



FIGURE 2.8: HRED model architecture (Serban et al., 2015)

### 2.3.1 Hierarchical Response Generation Models

One of the key improvements of the standard SEQ2SEQ model addresses the turn-taking nature of dialogue. Initially designed for machine translation, SEQ2SEQ assumed passing an utterance in the source language at the input, and producing its translation in the target language at the output. This approach was adopted for the response generation by 'flattening' the dialogue context in one large input sequence. Although better than providing the model with no context at all, flattening it has the obvious shortcomings that the information about the speakers gets lost, and the overall length of the input becomes extremely high thus resulting in additional challenges with model training and often requiring simply cutting the context quite severely. To address this, Serban et al. (2015) proposed the Hierarchical Recurrent Encoder-Decoder (HRED) model — see Figure 2.8.

In HRED, the encoding procedure is 2-stage: first, every utterance of the context is encoded with an utterance-level encoder RNN into a dense vector. Then, these compact latent utterance representations are encoded using another RNN, producing a dense vector similar to the one at the previous stage. This final dialogue context representation serves as the initial state of the decoder, equivalent to that in the standard SEQ2SEQ.

In addition to the reasons mentioned above, this hierarchical nature of the model also helps training the entire architecture more efficiently. Backpropagation through time (BPTT), the technique normally used to train recurrent networks, assumes passing gradients through the RNN's weights word-by-word throughout the entire input sequence, thus resulting in exponentially small gradient values propagated to the first encoding steps (also referred to as 'vanishing gradient'). The hierarchical architecture reduces the length of gradient paths in the computational graphs, thus

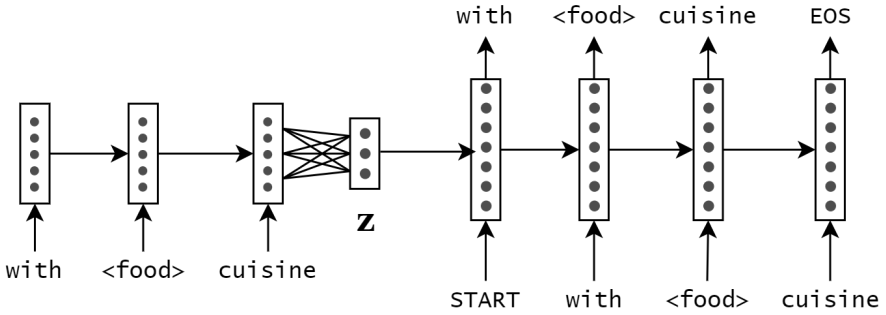



FIGURE 2.9: RNN-based autoencoder

alleviating this problem. This was especially relevant for 'plain' RNNs which widely suffer from vanishing gradient (see our discussion of RNNs in Section 2.1.4).

### 2.3.2 Representation Learning with Autoencoders

A key aspect of modelling dialogue is obtaining an efficient latent representation of the underlying utterances capturing their meaning and invariant to the surface variations. This *representation learning* problem is normally tackled with a type of models called *autoencoders*. Autoencoder (AE) is a model that is trained to attempt to copy its input to its output (Goodfellow et al., 2016). An autoencoder normally consists of two parts: the encoder and the decoder. The encoder produces a latent representation of the model's input, which then the decoder uses to produce the output, which is ideally the copy of the input. Crucially in this process, the resulting latent representation, i.e. the encoder's output (also referred to as the *bottleneck*) is of a dimensionality significantly lower than that of the input and the reconstructed output. By learning to 'compress' the data into a more compact representation (i.e. of a reduced dimensionality) which still contains enough information for the decoder to reconstruct the input, the autoencoder determines which 'features' of the input are informative and which ones are not and can be ignored. In Figure 2.9 is shown an autoencoder with both the encoder and the decoder being RNNs, a similar model to the one we will be working with in Chapter 7 — in the picture, $z$ is the latent input representation that is regarded as the main autoencoder's output.

The key question in training an autoencoder is to make its latent space *continuous*. That is, given a pair of latent vectors for the two inputs seen during training, an autoencoder with a continuous latent space would not only be able to reconstruct those two inputs, but also to produce meaningful reconstructions from arbitrary positions along the line between the encodings. The type of an autoencoder directly aimed at learning continuous latent spaces is based on the Bayesian methods and is called the Variational Autoencoder (VAE, Kingma and Ba, 2015; Rezende et al., 2014). VAE represents the encoding $z$ as a continuous latent variable produced by the probabilistic *recognition model* (the counterpart of AE's deterministic encoder) approximating the

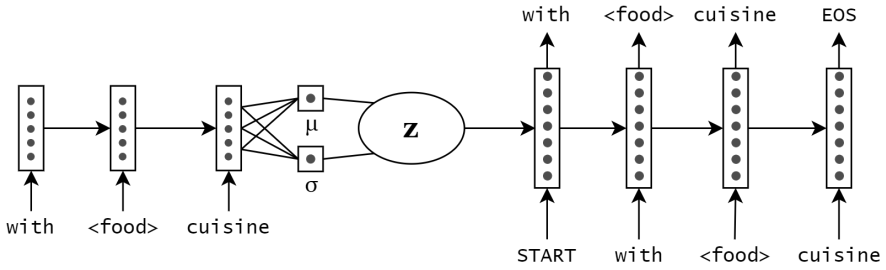



FIGURE 2.10: RNN-based VAE

posterior distribution of this latent variable given the input $q(z \mid x)$, in practice usually a diagonal Gaussian. The actual reconstruction is then obtained by running the *generation model* (the counterpart of AE's decoder) conditioned on a *sample* from $q(z \mid x)$.

The variational counterpart to the RNN autoencoder is shown in Figure 2.10 — the recognition model there consists of an RNN encoder and a linear projection layer generating parameters $\mu$ and $\sigma$ of the $q(z \mid x)$ Gaussian, a sample from which then goes into the generation model.

As opposed to the MLE optimisation objective of a regular AE, VAE is trained to optimise the Evidence Lower Bound (ELBO) objective which looks as follows:

$$\mathcal{L}(x) = -KL(q(z \mid x) \parallel p(z)) + \mathbb{E}_{q(z|x)} \log p(x \mid z) \leq \log p(x) \tag{2.12}$$

where $p(z)$ is the prior distribution of $z$ usually set to a standard Gaussian with $\mu = 0, \sigma = 1$, and $\log p(x)$ is the data likelihood, which AE optimises. The first term of the formula is Kullback-Leibler divergence between the approximate posterior and the prior:

$$KL(q \parallel p) = \sum_{x \in X} q(x) \log \frac{q(x)}{p(x)} \tag{2.13}$$

where $X$ is the probability space $q$ and $p$ are defined on.

KL divergence provides an expectation of the likelihood ratio between the two distributions (second term) with respect to $q(x)$. The intuition behind it is measuring how much more probable the data is under one distribution than under the other. In the VAE optimisation objective, the KL term keeps the model from making $q(z \mid x)$ deterministic (which it could become if the recognition network learned to output $\sigma$ close to 0) and instead forces it to be closer to the prior $p(z)$.

In practice, penalising the divergence between the prior and posterior comes with a challenge: early into the training, the KL term tends to reduce to zero, essentially making it identical to the prior. This problem is known as the 'vanishing KL term' problem (Bowman et al., 2016).

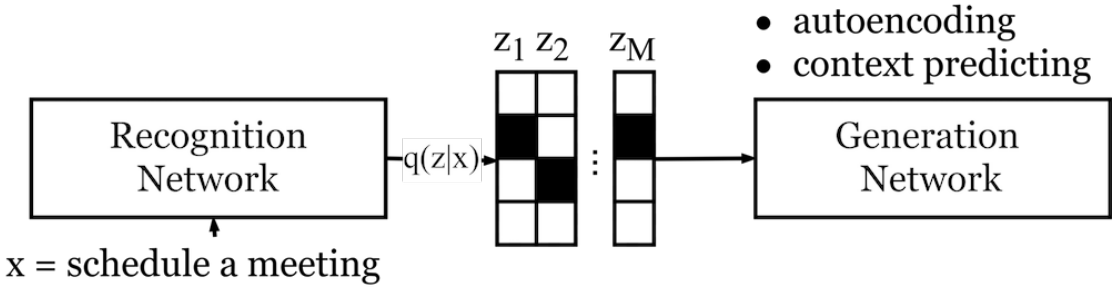



FIGURE 2.11: The DI-VAE/DI-VST models (Zhao et al., 2018)

In the context of the RNN-based models which are extremely prone to overfitting, the source of stochasticity (i.e. latent variable $z$) can be overfitted as well, so it does not matter what information is encoded in it as the network adapts itself to noise. Such a model would essentially converge to its non-variational and severely overfitted counterpart. Therefore, the optimisation of the ELBO objective is highly problematic in practice.

In order to facilitate learning, Bowman et al. (2016) propose the 'KL term annealing' technique. KL weight annealing imposes a weight upon the KL term that is set to zero at the beginning of training, effectively leaving the model with a regular MLE objective. After a certain number of epochs of initial fitting to data, the annealing weight starts growing up on the interval $[0, 1]$, eventually ending up with the objective as in Eq. 2.12. However, in practice, KL annealing is itself very unstable and needs careful fine-tuning of the annealing schedule. Therefore, more principled techniques for ELBO optimisation were introduced later on — the most notable is the approach of mutual information maximisation between the input and the latent variable (Zhao et al., 2019) and the version of it for discrete latent variables (Zhao et al., 2018). The latter is called Discrete Information VAE (DI-VAE) and is shown in Figure 2.11. It is a variant of a VAE with two modifications. Firstly, its optimisation objective accounts for the mutual information $I$ between the input and the latent variable which is found to be implicitly discouraged in the original VAE objective (see the authors' derivation in Eqs. 2.14 and 2.15).

$$\begin{aligned}\mathcal{L}_{VAE} &= \mathbb{E}_x \left[ \mathbb{E}_{q_{\mathcal{R}}(z|x)} \left[ \log p_{\mathcal{G}}(x \mid z) \right] - KL(q(z) \parallel p(z)) \right] \\ &= \mathbb{E}_{q(z|x)p(x)} \left[ \log p_{\mathcal{G}}(x \mid z) \right] - I(Z, X) - KL(q(z) \parallel p(z)),\end{aligned} \tag{2.14}$$

$$\begin{aligned}\mathcal{L}_{DI\text{-}VAE} &= \mathcal{L}_{VAE} + I(Z, X) \\ &= \mathbb{E}_{q_{\mathcal{R}}(z|x)p(x)} \left[ \log p_{\mathcal{G}}(x \mid z) \right] - KL(q(z) \parallel p(z))\end{aligned} \tag{2.15}$$

where $x$ is the input utterance, $z$ is the latent variable ($X$ and $Z$ corresponding to their batch-wise vectors), $\mathcal{R}$ and $\mathcal{G}$ are the recognition and generation models (implemented as RNNs)

respectively, and $q(z) = \mathbb{E}_x[q_{\mathcal{R}}(z \mid x)]$.

Secondly, the latent variable $z$ in DI-VAE is *discrete* as opposed to the continuous one in a regular VAE. The discrete latent code lends itself well to interpretation and can be used e.g. to cluster inputs by the values of the certain 'bits' of their latent codes. The discrete nature also makes the calculation of the KL term more tractable via the Batch Prior Regularisation technique (Zhao et al., 2018):

$$KL(q'(z) \parallel p(z)) = \sum_{k=1}^{K} q'(z = k) \log \frac{q'(z = k)}{p(z = k)} \tag{2.16}$$

where $K$ is the number of $z$'s possible values and $q'(z)$ is the approximation to $q(z)$ over $N$ data points:

$$q'(z) = \frac{1}{N} \sum_{n=1}^{N} q_{\mathcal{R}}(z \mid x_n) \tag{2.17}$$

In addition to DI-VAE, Zhao et al. (2018) introduced DI-VST, DI-VAE's counterpart working in a Variational Skip-Thought manner (Hill et al., 2016). Specifically, it reconstructs the input $x$'s previous ($x_p$) and next ($x_n$) context utterances with the corresponding independent generation models $\mathcal{G}^p$ and $\mathcal{G}^n$:

$$\mathcal{L}_{DI\text{-}VST} = \mathbb{E}_{q_{\mathcal{R}}(z|x)p(x)}[\log p^n_{\mathcal{G}}(x_n \mid z) p^p_{\mathcal{G}}(x_p \mid z)] - KL(q(z) \parallel p(z)) \tag{2.18}$$

We are going to use DI-VAE and DI-VST later in Chapter 4, as well as empirically compare them to the 'classic' VAE on a downstream dialogue response generation task.

### 2.3.3 Latent Variable Models for Dialogue Response Generation

Apart from resulting in a continuous latent space, the use of a latent variable has another intuitive benefit for dialogue — increasing the diversity of responses. In chat-oriented dialogue, low diversity and insufficient informativeness in a common problem with Seq2Seq models. Given the extremely high variance of responses given their contexts in the training corpora (more on those later in Section 2.7), the regular MLE optimisation objective of such models forces the

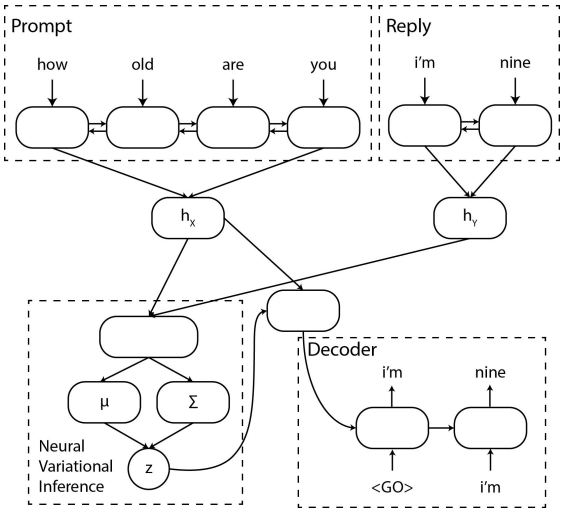



FIGURE 2.12: Latent variable SEQ2SEQ model (Cao and Clark, 2017)

model to learn the 'corpus-average', i.e. most recurring responses in the training set which usually are 'I don't know', 'I'm not sure' and the like (Jiang and de Rijke, 2018).

A series of approaches addressed that problem, e.g. Li et al. (2016a) used maximisation of mutual information (MMI) between the context and response as part of the objective function; Zhang et al. (2018) formulated it in terms of the Variational Information Maximisation Objective (VIMO) and approached the problem under the adversarial learning framework.

A widely used technique to increase diversity of responses was using random sampling in the SEQ2SEQ decoder (Ippolito et al., 2019), i.e. drawing a random entry from the model's probability distribution for every output token. But this often resulted in degraded generation performance as the MLE objective by definition only assumes optimising for the most probable output, and even the 2nd most probable word can be irrelevant or disfluent in most cases.

In turn, latent variable models allow to directly incorporate non-determinism into the model as well as into the training objective. From the dialogue perspective, it means that for every dialogue context, there can potentially be multiple correct responses, and the model will optimise for all of them. From the Bayesian optimisation point of view, that means learning the posterior distributions over these responses instead of point-wise optimisation in case of MLE, as we discussed with VAEs previously. Such models were introduced by Serban et al. (2017b) and Cao and Clark (2017) — the latter is visualised in Figure 2.12, we are going to discuss it below.

Under this approach, the output probability gets conditioned on (in addition to the input as in regular MLE training):

$$P(Y \mid X) = \int_z P(Y \mid z, X)P(z)dz \tag{2.19}$$

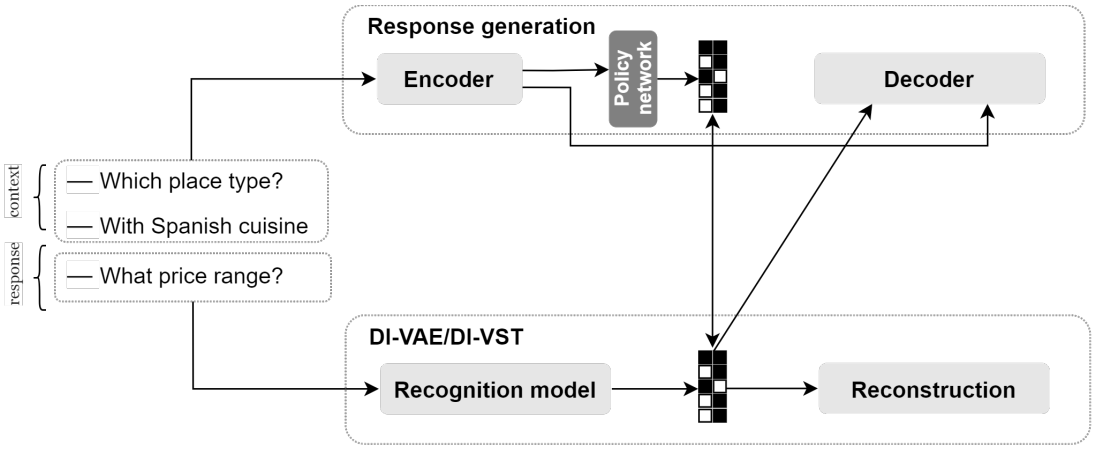



FIGURE 2.13: The LAED architecture (training)

where $P(z) = \mathcal{N}(0, I_n)$ is usually set to a standard Gaussian prior. This conditional probability denotes a distribution of correct responses for a given input mentioned above, and as the value for $z$ is sampled, the specific response from the distribution gets picked for further generation. With the presence of the system's output $Y$ different from the input $X$ (as opposed to the VAE case), the ELBO objective takes the following form:

$$\log P(Y \mid X) \geq -KL(Q(z \mid X, Y) \parallel P(z)) + \mathbb{E}_{z \sim Q} \log P(Y \mid z, X) \tag{2.20}$$

where $Q$ is the *proposal* (or *variational*) distribution which is used to approximate the posterior $P(z \mid X, Y)$ during the optimisation.

The Discrete-Information VAE technique we discussed previously in Section 2.3.2 was integrated into an encoder-decoder architecture, resulting in the Latent Action Encoder-Decoder (LAED) model (Zhao et al., 2018) — see Figure 2.13. The main part of LAED is the encoder-decoder SEQ2SEQ model which (1) encodes the dialogue context $c$ (including the user's last turn) with a hierarchical recurrent encoder and then produces an approximate posterior $p_\pi(z \mid c)$ of the latent variable $z$ using its 'policy' feed-forward network $\pi$, (2) decodes the system's response $y$ using samples from $p_\pi(z \mid c)$ and $c$.

Crucially though, at training time, it works in a multi-task setup with a DI-VAE/DI-VST model whose recognition model $\mathcal{R}$ (separate from the main model's encoder) encodes the *response* and approximates a posterior $q_\mathcal{R}(z \mid y)$. It then reconstructs either the response itself (DI-VAE), or its previous+next context utterances (DI-VST) using its generation model, also separate from the main task's decoder. During training, the main task's decoder uses samples from $q_\mathcal{R}(z \mid y)$ from the auxiliary DI-VAE/DI-VST instead of $p_\pi(z \mid c)$, and the policy $\pi$ only learns to reproduce $q_\mathcal{R}(z \mid y)$ via the MLE objective: $\mathbb{E}_{p(y|c)}\left[q_\mathcal{R}(z \mid y)\right]$.

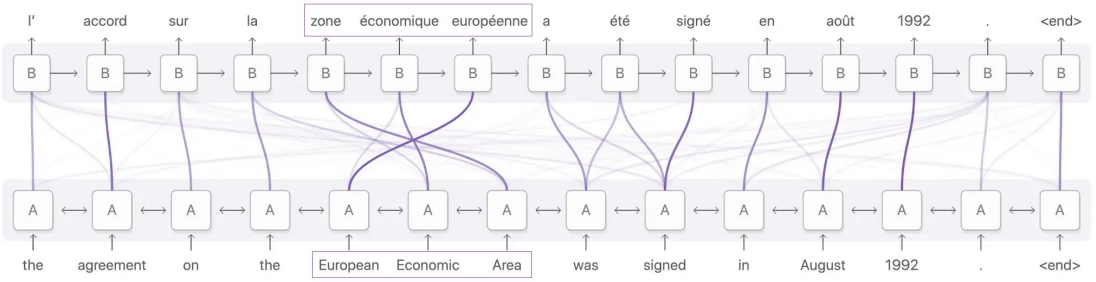



FIGURE 2.14: Visualisation of attention in machine translation (Olah and Carter, 2016)

LAED training objective is as follows:

$$\mathcal{L}_{LAED} = \mathbb{E}_{q_{\mathcal{R}}(z|y)p(c,y)} \left[ \log p_{\pi}(z \mid c) + \log p_{\mathcal{F}}(y \mid z, c) \right] \tag{2.21}$$

where $\mathcal{F}$ is the recurrent decoder. We are going to look at the potential of LAED's latent codes for dialogue knowledge transfer across datasets in Chapter 4.

### 2.3.4 Attention Mechanism

The most widely-used modification to the standard encoder-decoder architecture came, as the original model itself, from machine translation: the attention mechanism introduced by Bahdanau et al. (2015) addressed the problem of the insufficient information coming to the decoder from the final encoder state. While significantly more efficient than RNNs, LSTM/GRU models still do not produce the perfect representation of the input sequence for further decoding from it, and more importantly, this representation is static, while intuitively the translation takes place in segments.

For example, for generating one noun phrase the translating decoder in Figure 2.14 (Olah and Carter, 2016) would have to 'attend' mainly to this phrase at the input — that is, the translation of the phrase 'zone économique européenne' can be done without knowing its left or right contexts.

Attention is visualised in Figure 2.15: as opposed to the original SEQ2SEQ model only passing the final encoder state as the context for the decoder, this approach preserves all the intermediate encoder states and passes them all weighted with the corresponding *alignment scores* learned as part of the end-to-end training procedure. Alignment scores $\alpha_{t,i}$ tell how much information the decoder can infer from the encoder state $h_i$ (i.e. having encoded the tokens $x_1, ..., x_i$) while generating the $t$th output token.

$$\alpha_{t,i} = \frac{\exp(\text{score}(s_{t-1}, h_i))}{\sum_{i'=1}^{n} \exp(\text{score}(s_{t-1}, h_{i'}))} \tag{2.22}$$

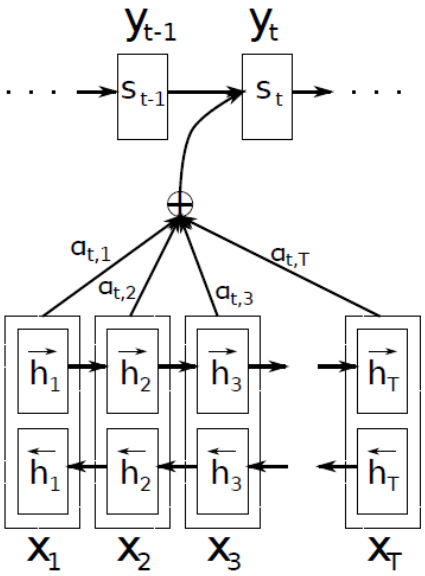



FIGURE 2.15: Attention mechanism (Bahdanau et al., 2015)

where $s_{t-1}$ is the previous decoder state, $h$ is the encoder state, and score is a scoring function implemented as a feed-forward neural network (among a series of other implementations) with trainable weights $W_a$ and $v_a$:

$$\text{score}(s_t, h_i) = {v_a}^{\mathsf{T}} \tanh\left(W_a[s_t; h_i]\right) \tag{2.23}$$

At the $t$th decoding step, the decoder receives the information from all the encoder states as a weighted mixture of the following form:

$$c_t = \sum_{i=1}^{n} = \alpha_{t,i} h_i \tag{2.24}$$

Luong et al. (2015) also explore a range of similar models. Specifically, *global* attention, the closest to that of Bahdanau et al. (2014), defines the alignment score for $t$th step via $s_t$ instead of $s_{t-1}$, as well as considering several variants of the score function:

$$\text{score}(s_t, h_i) = \begin{cases} s_t^{\mathsf{T}} h_i & \textit{dot} \\ s_t^{\mathsf{T}} W_a h_i & \textit{general} \\ {v_a}^{\mathsf{T}} \tanh\left(W_a[s_t; h_i]\right) & \textit{concat} \end{cases} \tag{2.25}$$

The alternative *local* attention, instead of calculating alignment scores of the decoder state with all the encoder's states, uses a fixed window $[p_t - D, p_t + D]$ where $p_t$ is the position within the

encoded sequence predicted by the model, and $D$ is a constant set empirically. The authors use 2 ways of calculating $p_t$: *monotonic* alignment where $p_t = t$, and *predictive* alignment defined as follows:

$$p_t = S \cdot \text{sigmoid}\left(v_p^\intercal \tanh(W_p h_t)\right) \tag{2.26}$$

where $W_p$ and $v_p$ are trainable parameters, and $S$ is the input sentence's length. In addition, in order to favour alignment scores around $p_t$, the authors place a Gaussian distribution centered around it on $\alpha$:

$$\alpha'_{t,i} = \alpha_{t,i} \exp\left(-\frac{(s - p_t)^2}{2\sigma^2}\right) \tag{2.27}$$

where $\sigma = \frac{D}{2}$.

Experimental evaluation on English-to-German machine translation task showed that local attention with predictive alignment was superior to other individual approaches, however the best result they obtained was with an ensemble of 8 models with different attention mechanisms and training aspects.

Although motivated by machine translation, attention was widely used for dialogue generation, both chat-oriented and goal-oriented. For the latter, this mechanism was in the core of another key technique —*copy-augmented decoding*.

### 2.3.5 Copy-Augmented Decoding

Copy-augmented Seq2Seq models (Eric and Manning, 2017; Zhao and Eskénazi, 2018) are used for addressing rare or out-of-vocabulary (OOV) words or some content of the user's query that needs to be reflected in the response as well, e.g. in the affirmation that all the details of the user's request are received, and the processing is started.

Attention-based copying is based on the idea of *pointer networks* (Vinyals et al., 2015) which produce a permutation of the input tokens as their output and were primarily aimed at solving the problem of sorting the input sequence and various combinatorial optimisation problems. In dialogue as well as other NLP tasks, hybrid models combining the 'pure generation' and the 'pure pointer' mechanisms are used, and here we will focus on Pointer-Generator Networks (See et al., 2017) shown in Figure 2.16.

Pointer-generator model defines the probability of generating the word $w$ at a certain decoding step as the following mixture:

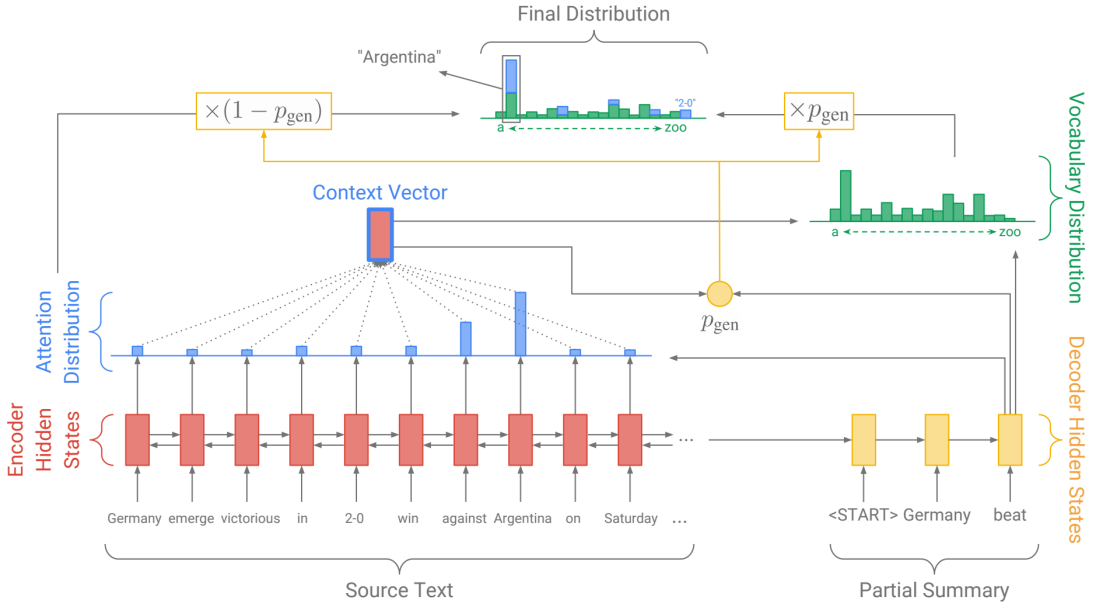



FIGURE 2.16: Pointer-Generator network architecture (See et al., 2017)

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_{t,i} \tag{2.28}$$

where $P_{vocab}$ is the probability of generating the word from the decoder's vocabulary under the conventional decoding procedure, and $a_i^t$ are the attention alignment scores of all the encoder states wherever the word $w$ is observed at the input. Finally, the mixture parameter $p_{gen}$ is defined as follows:

$$p_{gen} = \sigma\left(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}\right) \tag{2.29}$$

where $h_t^*$ is the attention context vector (equivalent to $c_t$ in the attention derivation above), $s_t$ is the decoder state, and $x_t$ is the decoder input — all at the decoding step $t$, and $w_{h^*}, w_s, w_x, b_{ptr}$ are trainable parameters.

In this approach, the decoder works with an 'extended vocabulary' which is the union of the original vocabulary and all the words appearing at the input. Therefore, for an OOV word, $P_{vocab}(w) = 0$, but if it is present at the input, it can still be transferred to the output if its attention weight is high. On the other hand, if this word is not present at the input, $\sum_{i:w_i=w} a_i^t = 0$, and it can only be generated based on the decoder's internal state as $P_{vocab}(w)$.

The particular implementation described above was used for hybrid abstractive/extractive document summarisation, but in general, the copy-augmented decoding technique is widely used in dialogue, semantic parsing (Jia and Liang, 2016), and language modelling (Merity et al., 2017).

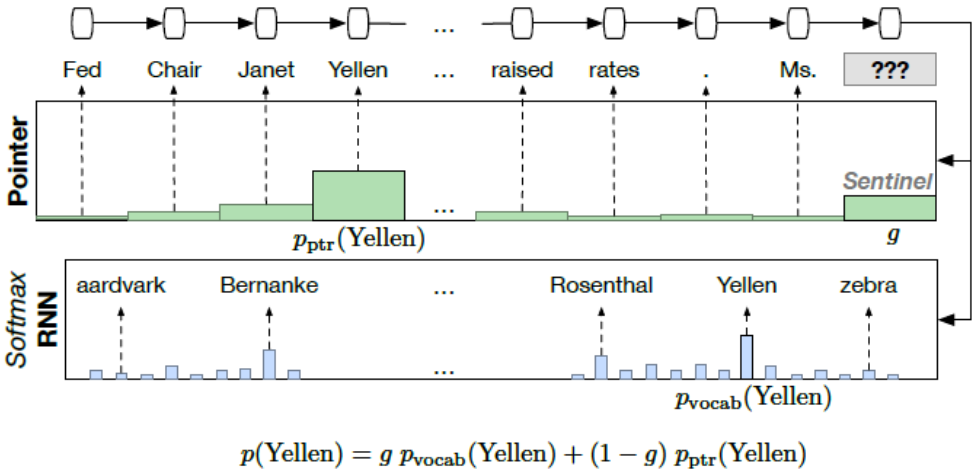



FIGURE 2.17: Pointer-Sentinel Mixture Model (Merity et al., 2017)

The latter — Pointer Sentinel Mixture Models (PSMs) (see Figure 2.17) — we are going to use in Chapter 4. As seen in the figure, PSM extends the pointer distribution with a *sentinel* element which is used to redistribute the probability mass in case of low confidence of the model's pointer part. The intuition is as follows: the lowest-confidence case of a 'vanilla' copy model — a near-uniform distribution derived from the attention scores — is supposed to be avoided in the PSM by putting the most of the probability mass on the sentinel. Then, this sentinel also working as a gating function in the hybrid pointer-generator prediction (see $g$ in Figure 2.17 and $p_{\text{gen}}$ in Eq. 2.28) — will control the model's final prediction. As such, $g$ can take values in the interval $[0, 1]$ representing the mixture parameter between the pointer and generator distributions, so that when $g = 0$, the next word is predicted from the pointer distribution only, and when $g = 1$, the model's confidence in pointer distribution is the lowest, and it falls back to generator-only prediction mode.

Apart from the enhancements of the SEQ2SEQ model, attention was also used as a key component of a text representation itself — specifically, of a hierarchical text representation model for document classification (Yang et al., 2016). Under their approach, two attention types were calculated — utterance-level attention representing the contribution of a specific word in the overall meaning of a sentence, and document-level attention representing a similar relation for individual sentences within the document. Later on, attention-based approach to text representation developed into a separate technique called *self-attention* which we are going to describe below.

### 2.3.6 Self-Attention

The original attention mechanism brought the idea of calculating the alignment scores between different parts of a sequential model's internal state. Self-attention introduced by Vaswani et al.

(2017) uses the idea of alignment as the main means of representing sequential input: instead of directional word-by-word encoding, self-attention assumes computing alignment scores of every word to every other one at the input, thus producing a representation of every word in the global context of the entire input sequence. More formally, the alignment scores are calculated via 3 quantities associated to every word $x_i$: 'key', 'query', and 'value' ($k_i$, $q_i$, $v_i$, respectively). The model obtains those via the following 3 respective projection matrices:

$$\begin{aligned} q_i &= x_i^\top W^Q \\ k_i &= x_i^\top W^K \\ v_i &= x_i^\top W^V \end{aligned} \tag{2.30}$$

The mathematical formulation of self-attention is largely similar to that of the original attention mechanism, with the main difference being that it is defined via $k_i$, $q_i$, and $v_i$. As such, the self-attention score of how a word $x_j$ affects the target word $x_i$ is calculated as follows:

$$\text{score}_{i,j} = \frac{q_i \cdot k_j}{\sqrt{d_k}} \tag{2.31}$$

where $d_k$ is the dimensionality of the key vectors. This operation is referred to by the authors as *scaled dot-product attention*.

The second step is similar to the original attention — all the self-attention scores for $x_i$ are softmax-normalised into alignment weights:

$$\alpha_{i,j} = \frac{\exp(\text{score}_{i,j})}{\sum_{k=1}^{n} \exp(\text{score}_{i,k})} \tag{2.32}$$

Finally, the final representation of a word $x_i$ is obtained as a mixture of all the input words' values $v_j$ weighted by their alignment weights given the target word:

$$z_i = \sum_{j=1}^{n} \alpha_{i,j} \cdot v_j \tag{2.33}$$

Self-attention representations of individual words $z_i$ are then fed into a feed-forward network. This architecture is then duplicated in the form of several sub-models, with separate matrices $W^Q$, $W^K$, and $W^V$. The final representations produced by these sub-models are referred to as *self-attention heads*. As reported in the original paper, these independently initialised and trained heads produce different representation 'subspaces' which may account for different linguistic phenomena, e.g. anaphoric links or syntactic dependencies.

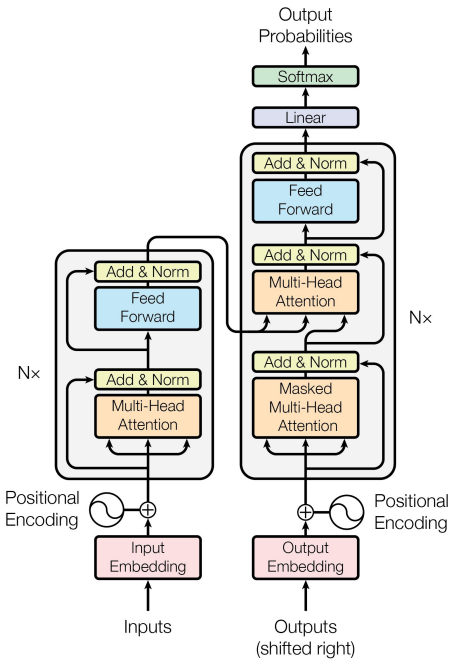



FIGURE 2.18: Transformer encoder-decoder architecture (Vaswani et al., 2017)

Self-attention mechanism lies in the core of the Transformer encoder-decoder model (shown in Figure 2.18). Both encoder and decoder of the Transformer use a similar logic to the one described above, with the encoder's output serving as the input representation and decoder's output being fed into the additional linear + softmax layers (together referred to as the 'language modelling head') used for generating the output probability distributions over the vocabulary tokens. More specifically, the decoder generates the output words token-by-token, just like the SEQ2SEQ model, but by attending to (1) the outputs of the stack of encoders and (2) the generated sequence up to the current timestep.

Transformers set the new state-of-the-art in a series of NLP tasks thus largely replacing RNN-based architectures as the main way of producing robust text representations. Self-attention architecture started a new generation of text models using the benefits of global-context word representation and increased parallelism of self-attention pipeline over recurrent sequential processing. The most notable of those models are Bidirectional Encoder Representations from Transformers (BERT) by Devlin et al. (2019) and Generative[3] Pretrained Transformers (GPT/GPT-2) by Radford et al. (2018) — trained on massive amounts of data and designed to be efficiently trained for a variety of downstream tasks, they largely enabled the transfer learning paradigm in NLP which will be discussed further.

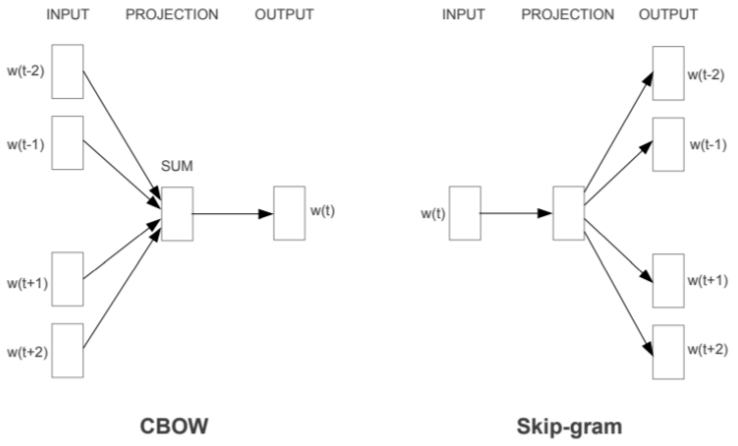



FIGURE 2.19: WORD2VEC model architecture (Mikolov et al., 2013)

## 2.4 Transfer Learning in NLP and Dialogue Modelling

Transfer learning is a direction in machine learning that assumes training one model for a specific task and then re-using the knowledge it has learned, partly or fully, on another task. The initial model is called the *base model*, and it is trained from large general-purpose datasets at the first stage called *pre-training*. At the second stage — *fine-tuning* — the base model (or its core part, e.g. utterance/dialogue encoder) is further trained for the target task, normally with additional task-specific parts introduced.

Transfer learning came to NLP from the Computer Vision (CV) community, where large-scale training of image recognition models became a common practice via IMAGENET, the visual recognition challenge and the corresponding dataset (Deng et al., 2009). State-of-the-art IMAGENET models, e.g. RESNET (He et al., 2016) and VGG (Simonyan and Zisserman, 2015) became widely used in the research community, and it soon became apparent that they can be adapted to a wide range of CV problems by fine-tuning to small in-domain datasets, eventually significantly outperforming the corresponding models trained from scratch.

### 2.4.1 Word Embedding Models

Similarly to CV with IMAGENET, NLP experienced a transformation with *word embeddings* which were the first widely transferable resources. Embeddings provide an efficient alternative to '1-hot' representation widely used before. 1-hot representation of a word assumes having a vocabulary-sized binary vector of zeros, with a single 'one' corresponding to the word's index in the vocabulary. Embeddings improved on that by representing words as real-valued vectors

[3]Here, "generative" is used in the sense of predicting (or generating) the next word given the context, i.e. language generation. We will also use this sense of the term "generative" later in Chapter 5.

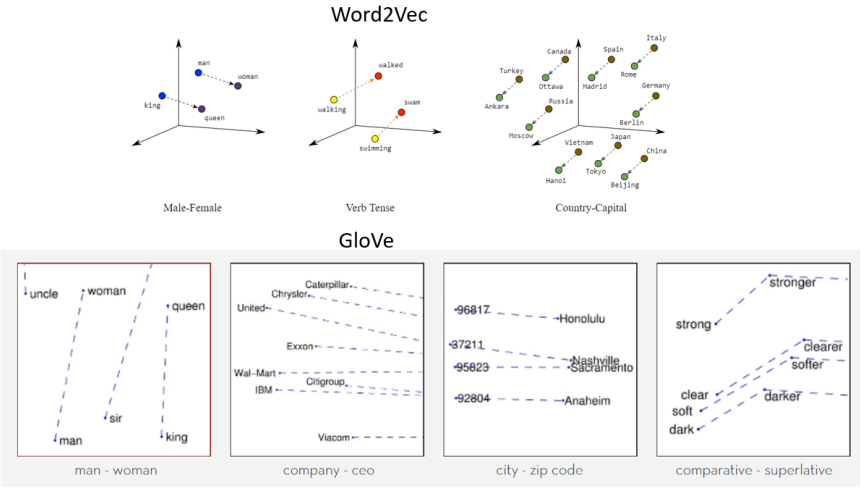



FIGURE 2.20: Example relations between WORD2VEC and GLOVE vectors

in a low-dimensional trainable space. Word embedding training objective is normally based on modelling a certain relationship between a word and its context (we will go into detail later) — this allows them to be trained in a 'self-supervised' fashion from internet-scale amounts of raw unannotated data.

In Figure 2.19 is shown WORD2VEC, one of the earliest and most widely-used word embedding models (Mikolov et al., 2013). The 2 versions of it: Continuous Bag-of-Words (CBoW) and Skip-gram — show that it can be trained either to predict the word from its surrounding context, or vice versa. The model's projection (or embedding) matrix that maps a 1-hot word vector into the latent space is the main result of the training.

Another early embedding model — GLOVE (Pennington et al., 2014) — uses word co-occurrence counts to predict the co-occurrence probability of a pair of words given their corresponding embedding vectors — the training objective makes the dot-product operation over a pair of word embedding vectors produce that probability. As multiple observations showed, both models' embedding spaces are able to encode different linguistic relations between words, e.g. 'male — female' or 'company — CEO', see Figure 2.20[4] for a visualisation.

Word representations produced by WORD2VEC and GLOVE gained extremely high popularity in the NLP community and were used to improve the performance of models in numerous tasks. Their principal shortcoming though was in the fact that they did not take word context while encoding it. That results in the inherent inability to encode sequences of words (e.g. utterances and paragraphs) — so that workaround approaches like 'mean-vector embedding' were used to represent phrases (i.e. the embedding of the phrase is the element-wise mean of the embeddings of individual words). Another problem is that contextless models cannot handle polysemous words like 'book', 'fly', 'like': the resulting representations of such words will correspond to

[4]Image credit: Renu Khandelwal (http://tiny.cc/w2v_glove_medium)

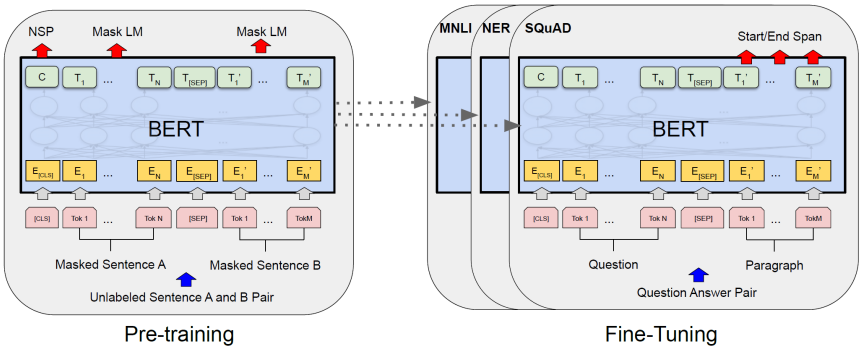



FIGURE 2.21: BERT 'pretrain-finetune' architecture (Devlin et al., 2019)

their most frequent sense in the dataset. The solution to these problems came with the next generation of embeddings based on sequence models.

### 2.4.2 Contextual Word Embeddings

Neural models for sequential data, e.g. LSTMs and Transformers brought the next generation of word embeddings. As such, there were introduced models like ELMo (Peters et al., 2018), ULMFiT (Howard and Ruder, 2018), and BERT Devlin et al. (2019). Trained as language models, they provided what is referred to as a 'contextual' representation (as opposed to the 'static' embeddings by WORD2VEC or GLOVE). Both models can be efficiently used in the downstream tasks by either replacing the 'language modelling head' with a task-specific one or training the two in a multi-task setup.

Among all the contextual embedding models, Transformer-based ones resulted in arguably the widest impact on the NLP community — specifically, BERT already mentioned above was designed for use in downstream tasks in a 'pretrain-finetune' fashion (shown in Figure 2.21). Specifically, this versatility is achieved by (1) organising the input in 2 parts, i.e. sentence A and sentence B (which may correspond to question/answer or context/response pairs in a downstream task) (2) pre-training in the multitask setup using the two objectives: Language Modelling (LM) and the Next Sentence Prediction (NSP). For the LM objective, BERT uses the notion of the Masked Language Model (MLM) which assumes predicting randomly chosen tokens of the input based on the full encoded representation of the input sequence, i.e. using both the left and the right contexts. The NSP task makes the model learn the relation between the 2 input sentences — in the base case, whether sentence B actually follows sentence A in the original text or it is a distractor. This secondary task helps improve the model's robustness to noise (if sentence B is set to be a randomly drawn distractor sentence) as well as make the model highly versatile in a variety of downstream classification-like tasks, e.g. Question Answering, Natural Language Inference, response retrieval and ranking.

Upon introduction, the original BERT achieved state-of-the-art results on 11 NLP tasks and was one of the most extensively used base models for text representation in transfer learning for NLP. More BERT variations followed up: for example, Robustly Optimized BERT approach (RoBERTa , Liu et al., 2019) trained with dynamic MLM masking, larger batch size, and disabled NSP task during pretraining; SpanBERT (Joshi et al., 2020) modifying MLM to mask and predict the entire spans of input instead of individual tokens; DistilBERT (Sanh et al., 2019), a version of BERT reduced in size by 40% while retaining 97% of the original model's accuracy, obtained via knowledge distillation.

### 2.4.3 Transfer Learning for Dialogue

Following wide success in fundamental NLP tasks (e.g. language modelling, semantic role labelling, coreference resolution), transfer learning techniques started emerging in dialogue.

The problem of dialogue system's domain adaptation was posed in Dialog State Tracking Challenge 3 which was focused on adapting a goal-oriented dialogue state tracker to a new domain using a small set of seed in-domain data (Williams et al., 2016) — that can be considered a transfer learning task, but from today's perspective, both pre-training and fine-tuning stages were bound to a very specific and narrow domain thus limiting the scale of the transfer. Nevertheless, several notable methods emerged from the challenge, all RNN-based ( Henderson et al., 2014d; Mrksic et al., 2015). Later, Williams et al. (2017) introduced the HCN model (described earlier in Section 2.1.2) that is designed for 2-stage training: the initial training stage from moderate amounts of data, and the consequent fine-tuning stage in an RL setup from interactions with real users or via an interactive human-in-the-loop process under the Conversation Learner framework (Williams and Liden, 2017).

More recently, a major shift in transfer learning for dialogue was brought by the Transformer model described above. Specifically, the introduction of Transformed-based GPT and GPT-2 (Radford et al., 2018) set the new state-of-the-art in human-like language generation. GPT models:

— are implemented using restricted unidirectional self-attention (where a token can only attend to its left context, as opposed to both left and right with e.g. BERT),

— consist of massive sets of parameters (the largest GPT-2 model has 1.5 billion parameters),

— are pre-trained at a large scale on WebText corpus consisting of millions of documents,

— set the new state-of-the-art on a number of language modelling tasks, while still underfitting on the WebText corpus, as reported by the authors.

GPT/GPT-2 employ language-modelling pre-training which allows them to be re-used in a number of generation tasks, including conversational response generation. They were rapidly adopted in the dialogue community, and a number of 'pretrain-finetune' approaches to dialogue followed, most notable of which are (1) TransferTransfo (Wolf et al., 2019), the winning submission at ConvAI2 challenge on persona-based chat-oriented dialogue as per the automatic metrics, (2) goal-oriented dialogue generation approach by Budzianowski and Vulic (2019), and (3) DialoGPT, an open-domain model by Zhang et al. (2020) pretrained at a large scale on the Reddit Conversations dataset. The latter achieved human-like performance in the 1-turn Turing test evaluation with humans. In Chapter 5, we are going to use GPT-2 as a base model in a 'pretrain-finetune' framework for dialogue domain adaptation, in both information-seeking and strictly task-oriented setups.

Models like ELMo, BERT, and GPT-2 showed that transfer learning can be applied in NLP as well as in vision. A number of NLP tasks have already experienced the benefits of transfer learning, and it is now considered among the best practices to approach new NLP problems via transferring those models' knowledge in a 'pretrain-finetune' fashion instead of training from scratch. In dialogue modelling, the benefits of transfer learning are starting to emerge so that conversation models become more human-like in naturalness, coherence, and appropriateness of their utterances — as well as in goal-oriented dialogue where those models need significantly less training data to start working with reasonable accuracy. In the next section, we are going to discuss the intuition of what aspects of dialogue can be transferred across domains and datasets.

### 2.4.4 Dialogue Transfer Learning Intuition: Lexical and Interactional Dialogue Similarity

With the variety of models and techniques of knowledge transfer presented above, it is important to have an actual intuition of why transfer learning applies to dialogue and what exactly can be transferred. There are two major aspects in which dialogues can vary, but nevertheless, lead to similar meanings: interactional and lexical. Interactional similarity is analogous to syntactic similarity — when two distinct sentences have effectively identical meaning — except that it occurs not only at the level of a single sentence, but at the dialogue or discourse level. Figure 2.22 shows examples of interactional variants that lead to very similar final contexts, in this case, that the user wants to buy an LG phone. These dialogues can be said to be effectively similar for this domain.

Lexical similarity, on the other hand, holds among utterances, or dialogues, when different words (or sequences of words) express meanings that are sufficiently similar in a particular domain (see again Figure 2.22). Unlike syntactic or interactional ones, lexical similarity is domain-specific — that is, a pair of dialogues which are equivalent sequences of dialogue acts, each one

| | | | | | |
|---|---|---|---|---|---|
| USR: | I would like an LG laptop sorry uhm phone | USR: | I would like a phone by LG. | SYS: | what would you like? |
| SYS: | okay. | SYS: | sorry a what? | USR: | an LG phone |
| | | USR: | a phone by LG. | SYS: | okay. |
| | | SYS: | okay. | | |
| SYS: | what would you like? | SYS: | you'd like a ...? | SYS: | so would you like a computer? |
| USR: | a phone | USR: | a phone | USR: | no, a phone. |
| SYS: | by which brand? | SYS: | by what brand? | SYS: | okay. by which brand? |
| USR: | LG | USR: | LG. | USR: | LG. |
| SYS: | okay | SYS: | okay | SYS: | okay. |

FIGURE 2.22: Interactional variations in a shopping domain

is represented with equivalent syntactic structures, are not considered lexically similar if one has to do with restaurant search and the other with booking movie tickets.

The intuition behind learning dialogue representations from data that are transferable across datasets and domains is capturing these similarities in a general 'dialogue footprint'. In this way, semantically similar dialogues with the same footprint would be clustered together — either by their embedding in the latent space or by explicit meaning representations e.g. derived from a semantic parser.

In case of linguistically informed models, Eshghi and Lemon (2014) developed a method similar to Kwiatkowski et al. (2013) for capturing lexical similarity by creating clusters of semantic representations based on observations that those clusters correspond to similar non-conversational actions observed within a domain (e.g. a database query, a flight booking, or any API call). Distributional methods could also be used for this purpose (Lewis and Steedman, 2013). In general, this kind of clustering is achieved when the domain-general semantics resulting from semantic parsing is grounded in a particular domain. We note that while interactional similarity in dialogue can be accounted for by semantic grammars or formal models of dialogue structure (such as DS-TTR, Eshghi et al., 2012 or KoS, Ginzburg, 2012), lexical similarity relations have to be learned from data.

## 2.5 Linguistically Informed Models of Dialogue

Linguistic resources are a major source of prior knowledge for dialogue models and in NLP tasks in general. In the setting of data-efficient training, where a model is limited in what it can learn examples, it is especially important to incorporate prior knowledge in it, in the form of e.g. an ontology or a grammar. In dialogue systems, there exist approaches making use of linguistic knowledge of various kinds — e.g. Ramachandran and Ratnaparkhi (2015) proposed a method to represent and track the dialogue state via Relational Trees built on top of Knowledge

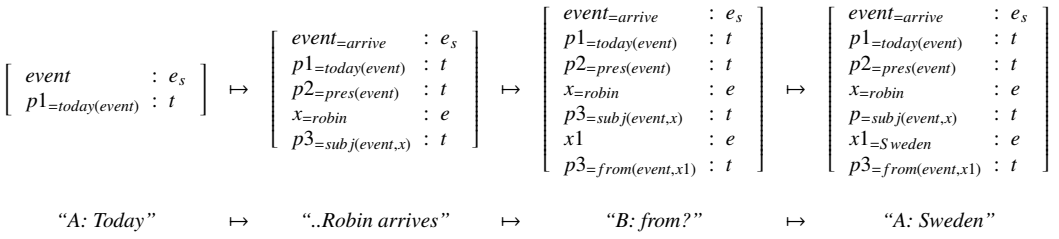



FIGURE 2.23: Incremental parsing with DS-TTR/DYLAN

Base entities extracted from the user's utterances; social dialogue model of Curry et al. (2018) benefited from the use of an ontology and entity linking as well.

In this section, we are going to give an overview of an approach to modelling dialogue entirely based on explicit linguistic representations — specifically, a formal semantic grammar — and discuss its applicability to low-resource dialogue system bootstrapping.

### 2.5.1 Dynamic Syntax and Type Theory with Records (DS-TTR)

Dynamic Syntax (DS) is an action-based, word-by-word incremental and semantic grammar formalism (Kempson et al., 2001; Cann et al., 2005), especially suited to the highly fragmentary and context-dependent nature of dialogue. This formalism is implemented in the incremental semantic parser for dialogue processing — DYLAN[5] (Eshghi et al., 2011; Eshghi, 2015; Purver et al., 2011). In DS, words are conditional actions — semantic updates, and dialogue is modelled as the interactive and incremental construction of contextual and semantic representations (Eshghi et al., 2015) — see Figure 2.23. The contextual representations provided by DS are fine-grained and jointly agreed upon by the interlocutors, as a result of processing questions and answers, clarification interaction, acceptances, self-/other-corrections, restarts, and other characteristic linguistic phenomena in dialogue — see Figure 2.24 for an example of how self-corrections and clarification requests are processed via a backtrack and search mechanism over the parse search graph (Hough, 2011; Hough and Purver, 2014a; Eshghi et al., 2015). Generation/surface realisation in DS is defined using trial-and-error parsing (see Section 2.5.2), guided by *a generation goal*, i.e. the semantic representation of the utterance to be generated. Generation thus proceeds word-by-word, similar to parsing (Purver et al., 2014; Hough, 2014).

Therefore, DS allows to track not only the semantic content of the current turn while it is being constructed word-by-word (during either parsing or generation) but also the context of the conversation as a whole, with the grounded/agreed content of the conversation encoded as well, see e.g. Figure 2.25 (Eshghi et al., 2015; Purver et al., 2010). Crucially for the dialogue model to be described below, the inherent incrementality of DS-TTR together with the word-level, as well

[5] DYLAN is derived from "Dynamics of Language"

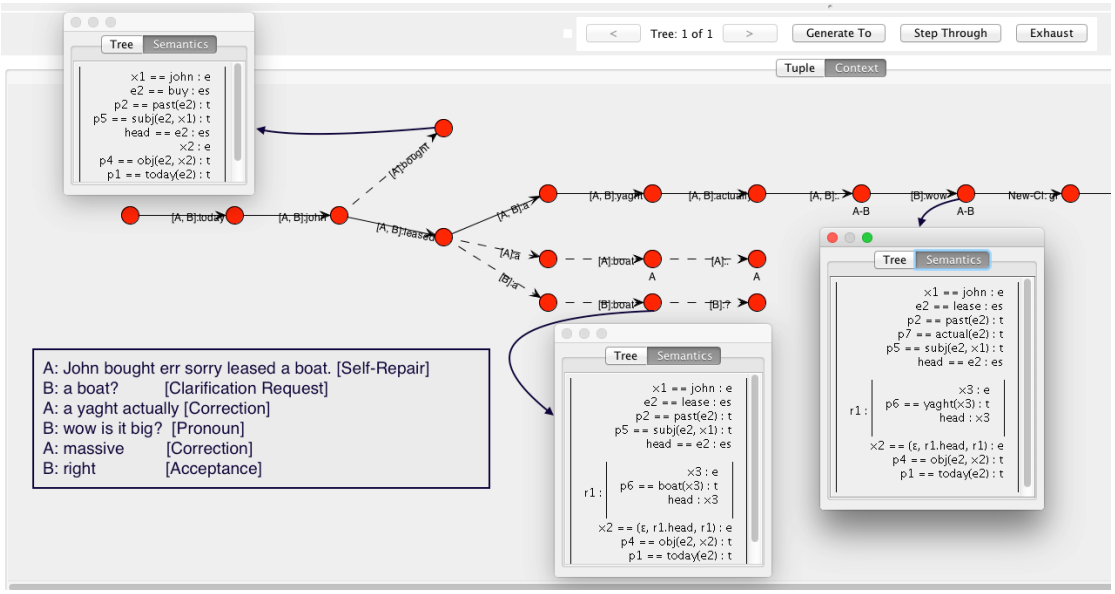



FIGURE 2.24: Processing self-corrections and clarification requests with DS-TTR/DYLAN

as cross-turn, parsing constraints it provides, enables word-by-word exploration of the space of grammatical dialogues, and the semantic and contextual representations that result from them.

Type Theory with Records (TTR) is an extension of standard type theory used in semantics and dialogue modelling (Cooper, 2005; Ginzburg, 2012). To support dialogue processing and allow for richer representations of the dialogue context, DS and the TTR framework were integrated (Purver et al., 2010; Purver et al., 2011; Eshghi et al., 2012). In TTR, logical forms are specified as *record types* (RTs), sequences of *fields* of the form $[\,l : T\,]$ containing a label $l$ and a type $T$. RTs can be witnessed (i.e. judged as true) by *records* of that type, where a record is a sequence of label-value pairs $[\,l = v\,]$, and $[\,l = v\,]$ is of type $[\,l : T\,]$ in case $v$ is of type $T$ (see Figure 2.23 for example record types).

A record type can be indefinitely extended, which naturally allows for representing incrementally growing meaning representations as more words are parsed or generated. That is, the semantics of the complete dialogue context expressed as an RT is a superset (or *supertype*) of the semantics midway in that same conversation. This is the key mechanism used in the dialogue system below. Our linguistically informed approach in Chapter 3 will be using this key DS-TTR property as well.

### 2.5.2 The BABBLE Dialogue Model

Incremental dialogue parsing described above combined with RL is the essence of the BABBLE model. Here, the parser acts as both NLU providing semantic dialogue state representation and word-by-word NLG by 'babbling' word sequences under the single shared grammar. RL, in turn, works as a trainable Dialogue Manager with word-by-word actions.

The two main resources for BABBLE are:

a) a DS-TTR parser *DS* — either learned from data (Eshghi et al., 2013a) or constructed by hand — for incremental language processing and more generally, for tracking the context of the dialogue using the model of feedback of Eshghi (2015), Eshghi et al. (2015, 2011);

b) a set $D$ of transcribed successful dialogues in the target domain.

In order to induce a fully incremental dialogue system from $D$, the following steps are performed:

1. Automatically induce the MDP state space, $S$, and the dialogue goal, $G_D$, from $D$;
2. Automatically define the state encoding function $F : C \rightarrow S$, where $s \in S$ is a binary state vector designed to extract from the current context of the dialogue, i.e. the semantic features observed in the example dialogues $D$; $c \in C$ is a DS context, i.e. a pair of TTR Record Types: $\langle c_p, c_g \rangle$ where $c_p$ is the content of the current clause (*pending*) as it is being constructed but not necessarily fully grounded yet; $c_g$ is the content already jointly built and *grounded* by the interlocutors (following the Dialogue Gameboard model of Ginzburg, 2012).
3. Define the MDP action set as the *DS* lexicon $L$ (i.e. actions are words);
4. Define the reward function $R$ as reaching $G_D$, while minimising dialogue length.

The generated MDP is then solved using RL, with a standard Q-learning method: train a policy $\pi : S \rightarrow L$, where $L$ is the DS Lexicon, and $S$ the state space induced using $F$. The system is trained in interaction with a semantic user simulation which is also automatically built from the dialogue data and described in the next section.

*The state encoding function F.* As shown in Figure 2.25 the MDP state is a binary vector of size $2 \times |\Phi|$, i.e. twice the number of the RT features. The first half of the state vector contains the grounded features (i.e. agreed by the participants) $\phi_i$, and the second half contains the current semantics built incrementally in the current dialogue utterance. Formally:

$$s = \langle F_1(c_p), \ldots, F_m(c_p), F_1(c_g), \ldots, F_m(c_g) \rangle \tag{2.34}$$

where $F_i(c) = 1$ if $c \sqsubseteq \phi_i$, and 0 otherwise.

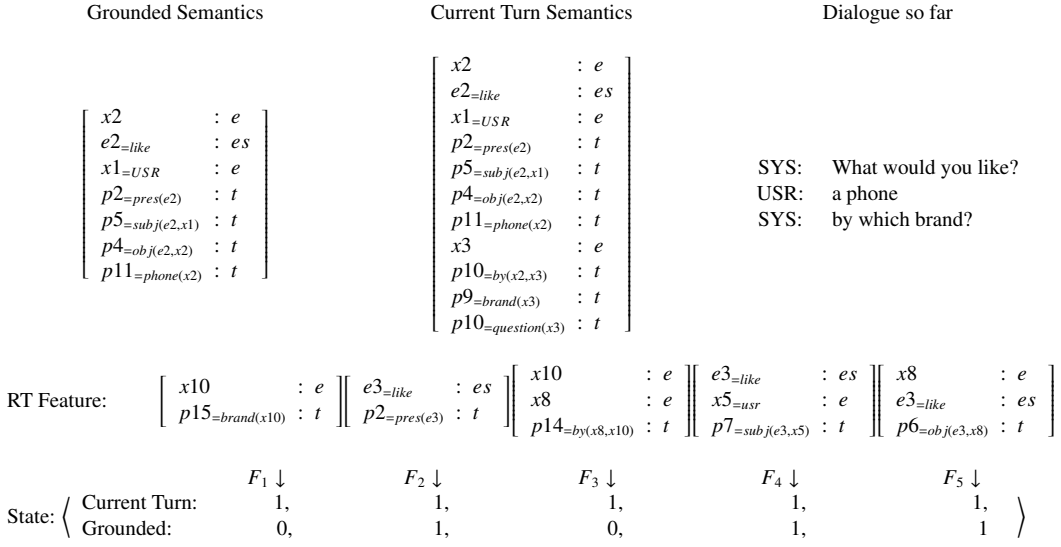



FIGURE 2.25: Semantics to MDP state encoding $F$ with RT features

### 2.5.3 Semantic User Simulation for BABBLE

An RL-based model, BABBLE requires a user simulator for efficient training. The simulator performs two key tasks during training: (1) generating user turns in the right dialogue contexts; and (2) word-by-word monitoring of the utterance so far generated by the system during exploration (i.e. babbling grammatical word sequences) by the system. Both tasks use the *DS* parser as well the state encoding function $F$ described above. They are thus performed based on the *semantic context* of the dialogue so far, as tracked by *DS*. The simulator is sensitive to the order in which information is received from the user since its context includes both the semantic features of the current turn and the history of the conversation.

The rules required for (1) & (2) are extracted automatically from the raw dialogue data $D$ using *DS* and $F$. The dialogues in $D$ are parsed and encoded using $F$ incrementally. For (1), all the states that trigger the user into action, $s_i = F(c)$ — where $c$ is a DS context prior to any user turn — are recorded and mapped to what the user utters in those contexts. For more than one training dialogue there may be more than one candidate (in the same context/state). The rules extracted in this way will be of the form:

$$s_{trig} \rightarrow \{u_1, \ldots, u_n\}$$

where $u_i$ are user turns.

The $s_i$'s prior to the user turns also immediately follow system turns. Therefore, in order to monitor the system's behaviour during training, one needs to check that the current state upon a a word generated by the system subsumes (or is extendible to) one of the $s_i$. This is performed through bitmask operation. The simulation can therefore semantically identify erroneous actions

(words) by the system. It would then terminate the learning episode and penalise the system immediately which considerably speeds up training.

To recap, the BABBLE model described above involves incrementally parsing dialogues and encoding the resulting semantics as state vectors in an MDP, which is then used for RL of word-level actions for system output (i.e. a combined incremental DM and NLG module for the resulting dialogue system).

## 2.6 Generalisation Power and Robustness of Dialogue Models

The key question in development machine learning models able to work with minimal amounts of training data is, how well they generalise to the data unseen during training. Specific to the dialogue systems we are going to work in this thesis, such novel properties of the data may be the details of the dialogue task, e.g. conversations coming from a different domain or containing slot types/values unseen in the training domains (see e.g. Zhao et al., 2017; Henderson et al., 2014b). It can also be some intrinsic property of the dataset itself, e.g. the presence of spoken disfluencies, out-of-domain (OOD) utterances, or just noise in the data — which can all be considered anomalous input. We then say that a dialogue system able to attain stable, consistently high performance across 'clean' data and that containing anomalous phenomena is *robust* to those. In this thesis, we are going to work with the following 2 types of robustness, categorised by the specific phenomena in the input data:

— *robustness to disfluencies* is concerned with the surface variations in the input utterances appearing due to the nature of spoken language (see the next section for a detailed problem description). A system robust to disfluencies is expected to attain similar performance on 'clean' data with those not present (e.g. examples collected in a controlled user study) as well as on more real-world conversations containing those phenomena (see a more detailed discussion in Section 2.6.1). We will explore this problem in Chapters 3 and 6

— *robustness to* OOD *input* represents a similar system's quality, with the phenomena of interest being user's input turns not belonging to the system's designated domain, e.g. 'put on my evening playlist' queried to a restaurant search system. We address this problem in a more specific way than the previous one, and so expect an OOD-robust system to be able to (1) correctly identify anomalous inputs in the dialogues, and (2) attain a performance level on OOD-containing data similar to that on purely in-domain (IND) dialogues. That is, the system is supposed to produce the originally designated responses for the IND turns as well as the special 'fallback' response for OOD turns signalising that it encountered anomalous input, with a minimal accuracy trade-off between the two (see an overview of the problem area in Section 2.6.2). We will explore this problem in Chapter 7.

### 2.6.1 Spoken Disfluencies and Data Efficiency

Humans process (parse and generate) language *incrementally* word by word, rather than turn by turn or sentence by sentence (Howes et al., 2010; Pickering et al., 1999; Ferreira et al., 2004). This leads to many characteristic phenomena in spontaneous dialogue that are difficult to capture in traditional linguistic approaches and are still largely ignored by dialogue system developers. These include various kinds of context-dependent fragments (Fernández and Ginzburg, 2002; Fernández, 2006; Kempson et al., 2017), false starts, suggested add-ons, barge-ins, and disfluencies.

In this thesis, we are interested in the following disfluencies: pauses, hesitations, false starts, and self-corrections — that are common in natural spoken dialogue. These proceed according to a well-established general structure with three phases (Shriberg, 1994):

$$\text{with } \underbrace{[\text{Italian}}_{reparandum} + \underbrace{\{uh\}}_{interregnum} \underbrace{\text{Spanish}]}_{repair} \text{ cuisine}$$

Specific disfluency structures have been shown to serve different purposes for both the speaker and the hearer (Brennan and Schober, 2001) — for example, a filled pause such as 'uhm' can elicit a completion from the interlocutor, but also serve as a turn-holding device; mid-sentence self-corrections are utilised to deal with the speaker's own error as early as possible, thus minimising effort.

In dialogue systems, the detection, processing, and integration of disfluency structures is crucial to understanding the interlocutor's intended meaning (i.e. robust NLU), but also for coordinating the flow of the interaction. Like dialogue processing in general, the detection and integration of disfluencies needs to be *strongly incremental*: it needs to proceed word by word, enabling downstream processing to begin as early as possible, leading to more efficient and more naturally interactive dialogue systems (Skantze and Hjalmarsson, 2010; Schlangen and Skantze, 2009).

Furthermore, incremental disfluency detection needs to proceed with minimal latency and commit to hypotheses as early as possible in order to avoid 'jittering' in the output and having to undo the downstream processes started based on erroneous hypotheses (Schlangen and Skantze, 2009; Hough and Purver, 2014b; Hough and Schlangen, 2015).

While many current data-driven dialogue systems tend to be trained end-to-end on natural data, they do not normally take the existence of disfluencies into account. The problem is that, taken together with the particular syntactic and semantic contexts in which they occur, disfluencies are very sparsely distributed, which leads to a large mismatch between the training data and actual real-world spontaneous user input to a deployed system. This suggests a more modular, pipelined approach, where disfluencies are detected and processed by a separate, domain-general

module, and only then any resulting representations are passed on for downstream processing. The upshot of such a modular approach would be a major advantage in generality, robustness, and data-efficiency.

#### 2.6.1.1 Incremental Disfluency Detection Models

Work on disfluency detection has a long history, going back to Charniak and Johnson (2001) who set the challenge. One of the important dividing lines through this work is the *incrementality* aspect, i.e. whether disfluency structure is predicted word by word.

In the non-incremental setting, as the problem is essentially sequence tagging, neural models have been widely used. As such, there are approaches using an encoder-decoder SEQ2SEQ model with attention (Wang et al., 2016) and a Stack-LSTM model working as a buffer of a transition-based parser (Wang et al., 2016; Wang et al., 2017b), the latter attaining superior results in the non-incremental setting.

Incremental, online processing of disfluencies is a more challenging task, if only because there is much less information available for tagging, i.e. only the context on the left. In a practical system, it also involves extra constraints and evaluation criteria such as minimal latency and revisions to past hypotheses which lead to 'jittering' in the output with all the dependent downstream processes having to be undone, thus impeding efficiency (Hough and Purver, 2014b; Purver et al., 2018).

Incremental disfluency detection models include Hough and Purver (2014b) who approach the problem information-theoretically, using local surprisal/entropy measures and a pipeline of classifiers for recognition of the various components of disfluency structure. While the model is very effective, it leaves one desiring a simpler alternative. This was made possible after the overall success of RNN-based models, which Hough and Schlangen (2015) exploit. In Chapter 6, we will build on top of this model, as well as evaluate it further.

Disfluency detection was also addressed in a multitask fashion by Schlangen and Hough (2017) whose secondary task is utterance segmentation — they demonstrate that the two tasks interact and thus are better approached jointly.

Language models have been extensively used for improving neural models' performance. For example, Peters et al. (2018) showed that a pre-trained language model improves RNN-based models' performance in a number of NLP tasks — either as the main feature representation for the downstream model, or as additional information in the form of a latent vector in the intermediate layers of complex models. The latter way was also employed by Peters et al. (2017) in the task of sequence labelling.

Finally, a multitask setup with language modelling as the second objective — the closest to our approach in Chapter 6 — was used by Rei (2017) to improve the performance of RNN-based Name Entity Recognition. The LM part of their model predicts both surrounding words of an input word which is done using a bidirectional LSTM (the forward one predicts the next word, and the backward one predicts the previous word). In our task, we can not make use of a backward model as we work in the word-by-word incremental fashion.

We note that there is no previous approach to multitask disfluency detection using a secondary task as general and versatile as language modelling. Furthermore, none of the works mentioned study how well their models *generalise* across datasets of different dialogue types, nor do they shed much light on what kinds of disfluency structure are harder to detect, and why.

### 2.6.2 Out-of-Domain Robustness and Data Efficiency

Data-driven approaches to dialogue systems development offered by the common bot building platforms (e.g. Google Dialogflow, Amazon Alexa Skills Kit, Microsoft Bot Framework) make it possible for a wide range of users to easily create dialogue systems with a limited amount of data in their domain of interest (e.g. restaurant search, travel booking, city info). Most task-oriented dialogue systems are built for a closed set of target domains, and in the setting of a low amount of in-domain training data, this leads to overfitting of machine learning methods and unpredictable performance outside their training sets. For a closed-domain dialogue system, it is extremely important to maintain predictable behaviour, and any failure to detect OOD utterances and respond with an appropriate fallback action[6] can lead to a frustrating user experience. In the setting of working with minimal training data, the latter is especially relevant since there is no access to 'real' OOD examples.

There have been a set of prior approaches for OOD detection which require both *in-domain* (IND) and OOD data (Nakano et al., 2011; Tur et al., 2014). However, it is a formidable task to collect sufficient data to cover in theory an unbounded variety of OOD utterances. In contrast, Lane et al. (2007) introduced an in-domain verification method that requires only IND utterances. Later, with the rise of deep neural networks, Ryu et al. (2017) proposed an autoencoder-based OOD detection method which surpasses prior approaches without access to OOD data. However, those approaches still have some restrictions such that there must be multiple sub-domains to learn utterance representation, and one must set a decision threshold for OOD detection. This can prohibit these methods from being used for most systems that focus on a single task. Moreover, recently, it was shown that density estimation models like autoencoders lack stability in telling between in-distribution and out-of-distribution data (Nalisnick et al., 2019) — and a standalone autoencoder does not suffice for our task, as we are going to

[6]See an example of the Alexa Skills Kit's built-in fallback action on Amazon Developer Blogs.

demonstrate empirically in Chapter 7. There, we will focus on studying the effect of OOD input on goal-oriented dialogue models' performance and propose a simple and efficient solution for improving their robustness only using IND data.

## 2.7 Dialogue Datasets and Data Collection

Deep learning methods described above perform at their best when provided with large amount of training data. In case of the NLP field in general, the main sources of general-purpose data are large-scale web resources: Wikipedia, online news resources, and posts on social networks (e.g. Reddit, Twitter). For dialogue, different kinds of datasets are used given the system type. Chat-oriented systems aimed at eliciting human-like open-domain conversation can be trained from large conversational (or conversation-like, e.g. comment threads on message boards) corpora. As such, the following datasets were used for training SEQ2SEQ conversation models (a wider review can be found in Serban et al., 2018):

— Cornell Movie Dialogs Corpus (Danescu-Niculescu-Mizil and Lee, 2011) — over 300,000 total utterances,

— OpenSubtitles (Lison and Tiedemann, 2016) — 400 million subtitle lines,

— Reddit conversations (Baumgartner et al., 2020) — over 3.7 billion comments,

— Twitter (Sordoni et al., 2015) — 29 million 'context-message-response' triples[7].

The intuition behind using movie subtitle corpora is that movie or TV series dialogues contain everyday conversations, and given enough coverage, it's theoretically possible to obtain open-domain chatting behaviour by mimic situations from the movies as well as learn to generalise over them to a certain degree.

In goal-oriented dialogue, the datasets used are more domain-specific. Some of the most widely-known are:

— 'Let's Go' (DSTC1, Williams et al., 2013) —15,000 dialogues in the bus information domain,

— Cambridge restaurants dataset (DSTC 2—3, Williams et al., 2016) with 3,000 dialogues in the domain of restaurant search,

[7]Larger datasets can be obtained from the Twitter Stream archives

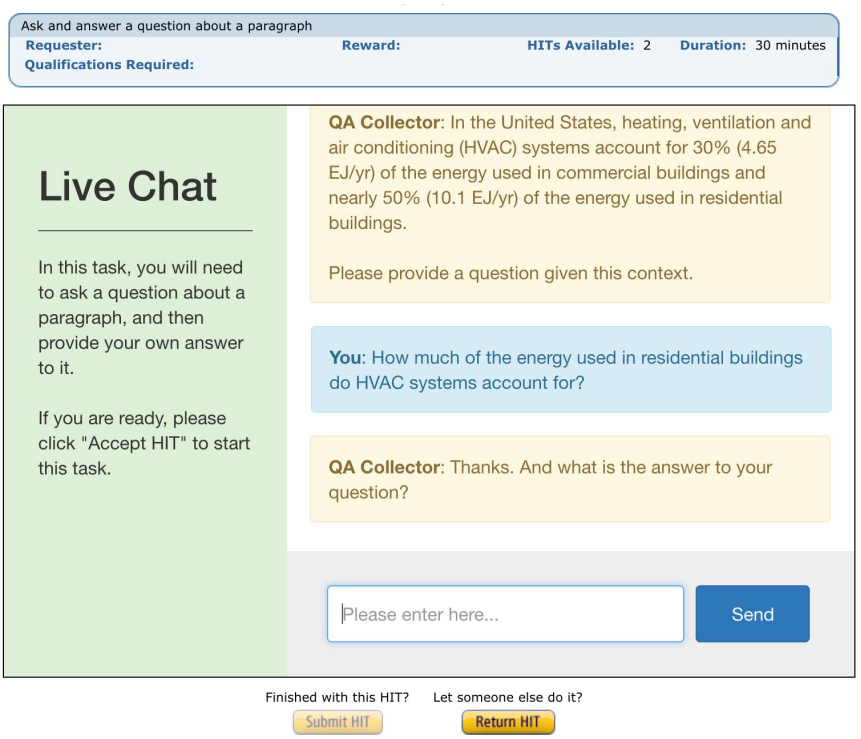



FIGURE 2.26: PARLAI web interface for Wizard-of-Oz data collection (Miller et al., 2017)

— Stanford Multi-Domain (SMD) dialogue dataset (Eric et al., 2017) with 3,000 dialogues in 3 goal-oriented domains, namely in-car navigation, weather information, and appointment scheduling,

— MULTIWOZ (Budzianowski et al., 2018, Eric et al., 2019, Zang et al., 2020) — a multi-domain, multi-task goal-oriented dataset with 10,000 dialogues. Domains represented in MULTIWOZ are restaurant, hotel, taxi, police, attraction, train, and hospital,

— FRAMES (Asri et al., 2017) with 1369 dialogues in the travel information domain addressing complex user's goals and more advanced real-world scenarios beyond linear form-filling.

— METALWOZ — the dataset collected for DSTC-8 Track 2 "Fast Domain Adaptation" (Lee et al., 2019a). with more than 37,000 human-human dialogues spanning the total of 227 tasks in 47 domains. The dialogues are collected in a way that human participants were assigned the role of bot or user, then given a problem domain and related specific task, and instructed to reach the user's goal over at least 10 dialogue turns.

Apart from the openly available testbeds, the data for domain-specific dialogue scenarios is normally collected via a technique called Wizard-of-Oz (WOz, Dahlbäck et al., 1998), where two humans interact with each other, one acting as the user and the other one simulating the behaviour of the potential dialogue system. Historically, this approach was used to conduct user experience studies, although in case of machine learning-based dialogue systems it is used as

the *seed* data for training the prototype system (also referred to as *bootstrapping*) for further fine-tuning from real interactions.

With training data being the principal asset in modern dialogue system development, it has become of key importance to incorporate data collection into the development pipeline in a principled way. Specifically, crowdsourcing platforms like Amazon Mechanical Turk (AMT)[8] and *Figure Eight*[9] gained wide adoption for collecting real-user data. Correspondingly, dialogue system frameworks and solutions introduced recently were designed with AMT integration in mind. For example, Rojas-Barahona et al. (2017) introduce their end-to-end trainable approach along with the WOz framework for data collection on AMT. Moreover, the ParlAI conversational platform (Miller et al., 2017) provides seamless AMT integration for WOz data collection as one of its key features — the web interface for WOz interactions is shown in Figure 2.26.

We observe that one of the key directions in conversational systems research is providing means to collect datasets of moderate amount in a principled way for rapid prototyping or bootstrapping dialogue systems. Still, the less are data needed for the system to perform reasonably well, the more flexible the system gets for use outside the academic testbeds. Therefore, advancing the training techniques for less dependence on data is of a high priority in dialogue systems research. In the next chapter, we are going to start our study on dialogue data efficiency by comparing two fundamentally different approaches to dialogue: linguistically informed models based on dialogue grammars and neural response retrieval models (discussed in Sections 2.5 and 2.2.1 of this chapter, respectively).

[8] https://www.mturk.com
[9] https://www.figure-eight.com

# Chapter 3

# Linguistic Knowledge or Learning from Examples: A Data Efficiency Perspective

We are going to start our research with an experimental study of dialogue models' generalisation power and robustness in the low-resource setup, i.e. in the task of bootstrapping a dialogue system from seed data. We perform the experiments in a controlled environment using the bAbI Dialog Tasks dataset (Bordes et al., 2017) and focus our attention on two fundamentally different types of models: a neural retrieval-based model MemN2N (Sukhbaatar et al. (2015), discussed in the previous chapter, and a linguistically informed model based on a semantic parser/generator DyLan. We look at their performance in the limited data setup with bAbI as well as their generalisation potential to more diverse and challenging input — for that, we introduce bAbI+[1], an augmented version of the bAbI dataset with increased surface complexity represented by simulated spoken disfluencies.

## 3.1 Motivation

Every practical machine learning model represents a trade-off between what is *learned* from data and what is *given* in the form of inductive biases. The range of possible inductive biases spans from the fundamental ones, e.g. specific network architectures tailored to different types of input data (CNNs for image input, RNNs for sequential input) to task-specific objective functions that those networks minimise. In dialogue, as well as in NLP in general, the main sources of inductive biases are linguistic resources, e.g. gazetteers, ontologies, thesauri, and knowledge

[1] Available at https://bit.ly/babi_plus

bases. Now, in the setting of minimal training data, the extent of what can be learned from it is limited very severely, so those linguistic resources and the corresponding inductive biases become the main way to obtain a generalisable model.

In the dialogue systems area, there are currently several key problems for the practical data-driven development of task-oriented systems, among them: (1) large amounts of dialogue data are needed, i.e. thousands of examples in a domain; (2) this data is usually required to be annotated with task-specific semantic information for the domain (e.g. various dialogue act schemes); and (3) the resulting systems are usually trained from the data that do not properly represent many characteristic phenomena of dialogue such as spoken disfluencies.

In overcoming issue (2), a recent advance in research on chat-oriented dialogue was the development of end-to-end systems, in which all components are trained from textual dialogue examples, e.g. Sordoni et al. (2015), Vinyals and Le (2015). However, as Bordes et al. (2017) argued, these end-to-end methods may not transfer well to task-based settings (where the user is trying to achieve a domain goal, such as booking a flight or finding a restaurant, resulting in an API call). They then presented an end-to-end method using Memory Networks (MemN2Ns), which achieves 100% performance on a testset of 1000 dialogues, after being trained on 1000 training dialogues. This method processes dialogues turn-by-turn, and so does not have the advantages of more natural incremental systems (Aist et al., 2007; Skantze and Hjalmarsson, 2010); nor does it really perform language generation, rather it is based on a retrieval model that selects from a set of candidate system responses seen in the data.

In this chapter, we will investigate two fundamentally different approaches: (1) MemN2N, a neural retrieval-based model of Sukhbaatar et al. (2015), and (2) a linguistically informed model that uses an incremental semantic parser/generator for dialogue based around the Dynamic Syntax grammar formalism — described in Sections 2.2.1 and 2.5.2, respectively. Specifically, we explore how well these methods overcome the requirement for large amounts (i.e. thousands of dialogues in a domain) of annotated dialogue data by putting them in a setup with up to 5 example dialogues from bAbI, Task 1. Then, in order to evaluate the systems' robustness to the unseen but highly likely variations in the user's input, we also introduce an extended, incremental version of the bAbI dataset, which we call bAbI+ (see Section 3.3): it adds some characteristic phenomena of spoken language — such as mid-utterance self-corrections — to the bAbI dialogues. Using this, we further experiment with the two systems and see how they are able process this more challenging data.

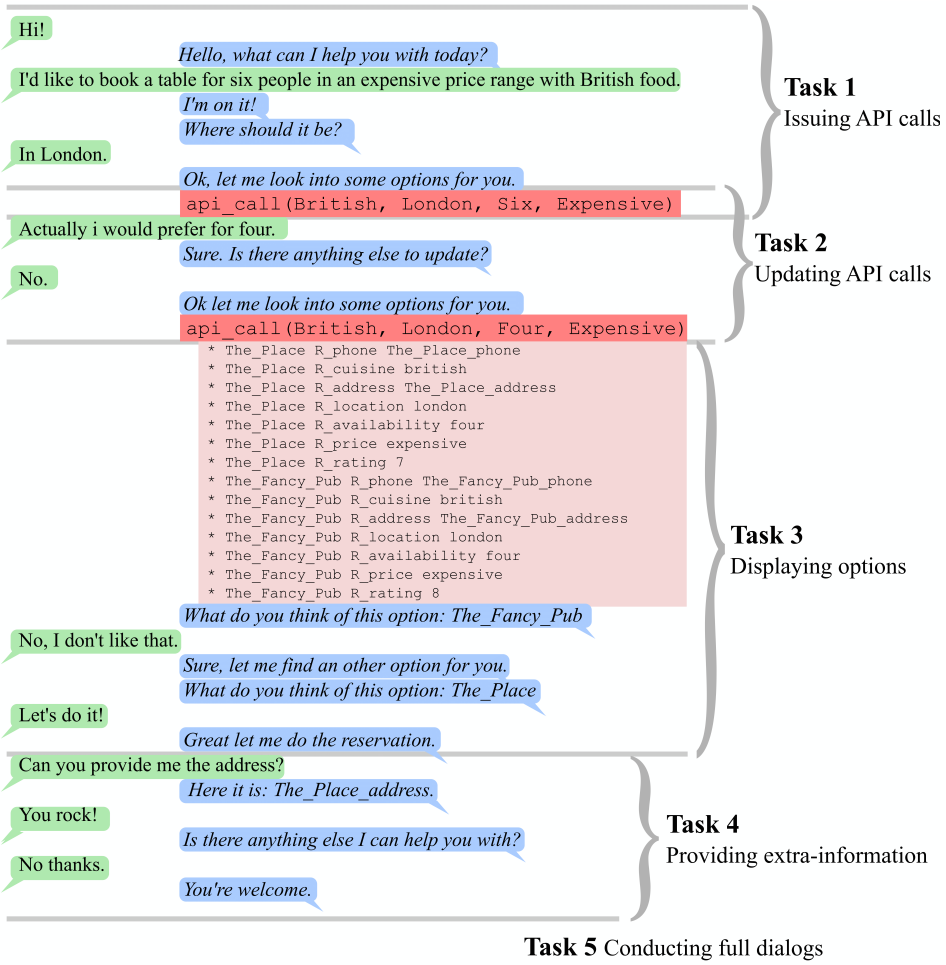



FIGURE 3.1: bAbI Dialog Tasks structure — shown are the synthetic Task1—Task 5 (Bordes et al., 2017)

## 3.2 Experimental Framework — bAbI Dialog Tasks

In this chapter, our focus is not on building dialogue systems, but on: **(1)** studying and quantifying the interactional and structural generalisation power of the DS-TTR grammar formalism and that of symbolic, grammar-based approaches to language processing more generally. We focus here on specific dialogue phenomena, such as mid-sentence self-corrections, hesitations, and restarts (see below); **(2)** doing the same for Bordes et al.'s response retrieval model MEMN2N, without the use of linguistic knowledge of any form; and **(3)** comparing (1) and (2).

In order to test and quantify the interactional and structural generalisation power of the two models, we need contrasting dialogue datasets that control for interactional vs lexical/syntactic variations in the input dialogues. Furthermore, to make our results comparable to the existing approach of Bordes et al. (2017), we need to use the same dataset that they have used. We therefore use Facebook AI Research's bAbI Dialog Tasks dataset (Bordes et al., 2017). These are goal-oriented dialogues in the domain of restaurant search. In the dataset, there are 6 tasks of

increasing complexity ranging from only collecting the user's preferences on restaurant and up to conducting full dialogues with changes in the user's goal and providing extra information upon request — see Figure 3.1 for an illustration. The first 5 tasks are 'clean' dialogues composed synthetically and they thus lack the features of natural everyday conversations. Task 6 (not shown in the figure) is the natural counterpart of the Task 5, containing dialogues with human users from the Dialog State Tracking Challenge 2.

After the original Bordes et al.'s result on Task 1, several studies have shown different ways in which MemN2Ns are outperformed: Eric and Manning (2017) introduced the Copy-Augmented Sequence-to-Sequence model that outperforms the MemN2N on Task 6; Williams et al. (2017) presented Hybrid Code Networks (discussed in the previous chapter), a combined RNN/rule-based model trainable in a 2-stage supervised + reinforcement learning setup, outperforming the MemN2N on Tasks 5 and 6.

However, none of these studies control for *the type of complexity* that might result in worse performance, and thus do not shed any light on why a particular architecture such as MemN2N might be at a disadvantage. While Task 5 dialogues have the full task complexity, conducting full dialogues with an unfixed user goal and additional information requests, they are still composed programmatically and contain minimal surface variation. The Task 6 dialogues on the other hand are complex both in terms of the surface variation and the task itself.

In order to study the specific effects of incremental variations in dialogue such as conversational disfluencies, we focus on Task 1, where in each dialogue the system asks the user about their preferences for the properties of a restaurant, and each dialogue results in an API call containing values of each slot obtained (e.g. `food-type=french`) — the ability of predicting the API calls correctly thus provides a direct measure of how well a particular model can interpret the dialogues. We would like to point out that we will be using the synthetic part of the bAbI Dialog Tasks dataset as a controlled experimental environment, and in the next section, we are going to present our modifications that we apply to this dataset in a programmatic way — similar to how the initial corpus was created — in order to simulate certain linguistic phenomena of interest.

## 3.3 The bAbI+ Dataset

The original bAbI dialogues were synthesised in a way that their main source of complexity is the dialogue goal itself, with its challengingness increasing from Task 1 to Task 5. In addition, they also contain some basic lexical/syntactic variation, e.g. "may i have a table with <cuisine> cuisine in a <price> price range in <place>?", "can you make a restaurant reservation with <cuisine> cuisine in a <price> price range in <location>?" However, Task 1 dialogues significantly lack any simulation of incremental and interactional variations vital for real-life

dialogues. In order to obtain such variation while keeping the controllable environment close to the laboratory conditions that bAbI offers, we created the bAbI+ dataset by systematically transforming the original dataset's dialogues.

bAbI+ is an extension of the bAbI Task 1 dialogues with disfluent dialogue phenomena (hesitations, restarts, and corrections — see below). This extension can be seen as orthogonal to the increasing task complexity which Tasks 2—5 offer: we instead increase the complexity of surface forms of dialogue utterances, while keeping every other aspect of the task fixed.

Our modifications model the disfluencies and communication problems in everyday spoken interaction in real-world environments. These variations are:

— *Hesitations*, e.g. as in "we will be `uhm` eight";

— *Restarts*, e.g. "can you make a restaurant `uhm yeah can you make a restaurant` reservation for four people with french cuisine in a moderate price range";

— *Corrections* affecting task-specific information – both short-distance ones correcting one token, e.g. "with french `oh no spanish` food", and long-distance NP/PP-level corrections, e.g. "with french food `uhm sorry with spanish food`", all within a single user utterance, rather than across multiple turns.

The phenomena above are mixed in probabilistically (with the aim to reflect the statistics from Hough, 2014) from the fixed sets of templates to the original data[2]. The modifications affect a total of 11,336 utterances in the 3998 dialogues. Around 21% of user turns contain corrections, 40% hesitations, and 5% restarts (they are not mutually exclusive, so that an utterance can contain up to 3 modifications). Our modifications, with respect to corrections in particular, are more conservative than those observed in real-world data: Hough (2014) reports that self-corrections appear in 20% of all turns natural conversations from British National Corpus, and in 40% of turns in the Map Task, a corpus of human-human goal-oriented dialogues.

Here is part of an example dialogue in the bAbI+ corpus showing some of the augmentations ( hesitations and corrections ) in the user's turns:

[2]See https://github.com/ishalyminov/babi_tools

**sys:** hello what can I help you with today?
**usr:** I'd like to book a uhm yeah I'd like to book a table
in a expensive price range no sorry in a cheap price range
**sys:** I'm on it. Any preference on a type of cuisine?
**usr:** with indian food no sorry with spanish food
**sys:** where should it be?
**usr:** in tokyo
**sys:** how many people would be in your party?
**usr:** for uhm yeah for eight people please
**sys:** ok let me look into some options for you

## 3.4 Models

### 3.4.1 MemN2N

We follow Bordes et al.'s setup by using a MemN2N (we took an open source Tensorflow implementation for bAbI QA tasks and modified it[3] according to their setup — see details below). For the description of the model architecture, please see Chapter 2. We're initially interested in the single-hop configuration with which Bordes et al. achieve perfect accuracy on bAbI Task 1.

In order to adapt the data for MemN2N, we transform the dialogues into <`story, question, answer`> triplets. The number of triplets for a single dialogue is equal to the number of the system's turns, and in each triplet, the `answer` is the current system's turn, the `question` is the user's turn preceding it, and the `story` is a list of all the previous turns from both sides. Other than that, each sentence in the `story` gets 2 additional tokens: the number of the turn, and the ID of the speaker (Bordes et al., 2017).

We also use the single embedding matrix $A$ for both input memories and the user's question; the same matrix is used for the output memories representation — in that we follow Bordes et al. (2017), and it corresponds to the "Adjacent" weight tying model in Sukhbaatar et al. (2015).

In our setup, there are no out-of-vocabulary words for the model during both training and testing, and for both bAbI and bAbI+ with the maximum sentence length taking account of the increase due to the transformations in bAbI+.

We train our MemN2Ns with an SGD optimiser for 100 epochs with a learning rate of 0.01 and a batch size of 8 — in this we again follow the configuration reported by Bordes et al. (2017) to be the best for bAbI Task 1.

[3] See https://github.com/ishalyminov/memn2n

### 3.4.2 DyLan: bAbI and bAbI+ Setup Details

Although DyLan's Dynamic Syntax grammar is learnable from data, the existing learned models in the prior work (Eshghi et al., 2013a; Eshghi et al., 2013b), were induced from a corpus of child-directed utterances, and there were some constructions as well as individual words that the resulting lexicons did not include. We therefore extended this induced grammar manually to cover the bAbI dataset, which, despite not being very diverse, contains a wide range of complex grammatical constructions, such as long sequences of prepositional phrases, adjuncts, short answers to yes/no and wh-questions, appositions of NPs, causative verbs etc — and all of this within and across dialogue turns/speakers. Using DyLan, we parsed all dialogues in the bAbI train and test sets, as well as on the bAbI+ corpus word-by-word, including both user and system utterances, in context. The grammar parses 100% of the dialogues, i.e. it does not fail on any word in any of the dialogues.

Our aim here is to assess the ability of a DyLan-based system to generalise from small data and compare this to the results of the MemN2N in Bordes et al. (2017). The latter is based on the retrieval of system responses given the dialogue history up to that point. Therefore, for direct comparison, and for simplicity of exposition, we set up an experimental testbed extending the semantic parser in the following way: we employ the logic originally presented as the BABBLE user simulation (Section 2.5.3), this time for the *system side*, resulting in a 'system simulation'. We then use this to predict a system response, by parsing and encoding the containing test dialogue up to the point immediately prior to the system turn. This results in a triggering state $s_{trig}$, which is then used as the key to look up the system's response from the rules constructed as per Section 2.5.3. The returned response is then parsed word-by-word as normal, and this same process continues for the rest of the dialogue. This method uses the full machinery of DS-TTR and our state-encoding method and will thus reflect the generalisation properties that we are interested in.

Our overall method described in Section 2.5.3 respects the turn ordering encountered in the data, or more generally the order in which semantic increments are added to context. This is because states are composed not only of the semantic features of the current turn, but also those of the conversation history. And thus they capture *the contextual boundary* at which a user turn is being generated or a system turn monitored (e.g. in the bAbI 'restaurant search' domain, a state might capture the fact that the user has already provided the cuisine type and the location of the restaurant).

Unlike many other approaches to goal-oriented dialogue, DyLan-based approach is not based on dialogue acts, and is word-by-word incremental. This means that there is no given/prior definition of what sequences of words or semantic updates constitute a dialogue turn: the system needs to learn this (or, in general, automatically construct from the data), or else it would just go

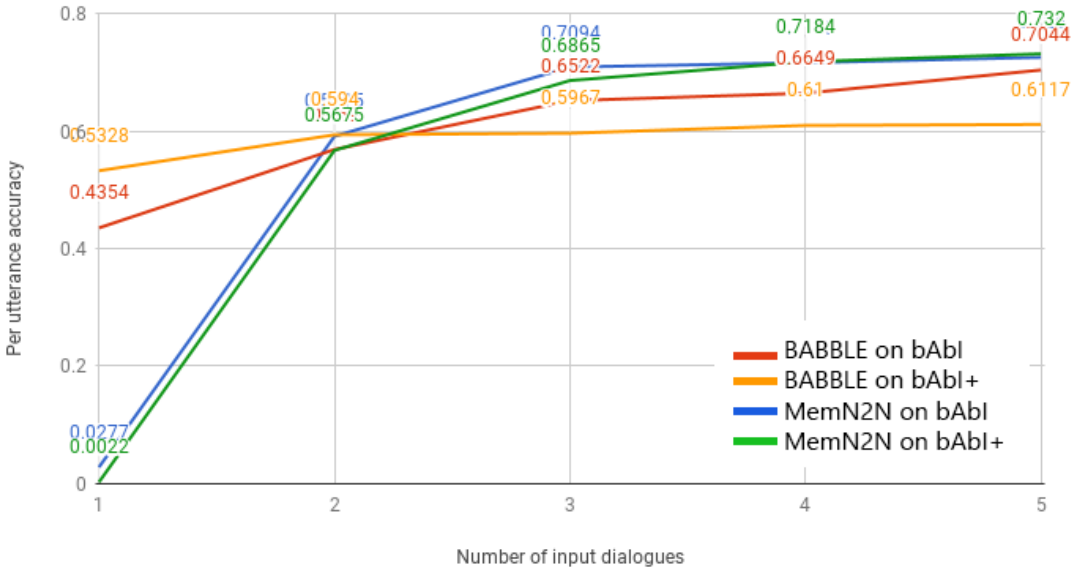



FIGURE 3.2: Few-shot performance of DyLan and MemN2N

on generating (i.e. exploring grammatical word outputs) without ever stopping. We prevent this by having the simulator interrupt the system at the *semantic* turn and clause boundaries that are encountered in the data; and thus the behaviour of the simulator determines the system's turn boundaries.

## 3.5 Experiments

### 3.5.1 Experiment 1: Generalisation from Small Data

We have now set out all we need to perform the first experiment. Since we are here interested in both (1) data efficiency and (2) robustness, we use all the bAbI and bAbI+ data — the train, dev, and test sets — in the cross-validation setup as follows: we train the MemN2N as well as construct DyLan's semantic context-response mapping from 1—5 examples selected at random from *the longest dialogues* in bAbI (note bAbI+ data is never used for training in this experiment). This process is repeated across 10 folds. The models are then tested on sets of 1000 examples selected at random, in each fold. Both the training and test sets constructed in this way are kept constant in each fold across DyLan and MemN2N. The test sets are selected either exclusively from bAbI or exclusively from bAbI+.

#### 3.5.1.1 Results: Predicting System Turns

Figure 3.2 shows per-utterance accuracies for the DyLan and MemN2N models. Per-utterance accuracy is the percentage of all system turns in the test dialogues that were correctly predicted. The table shows that DyLan can generalise to 74% of bAbI and 65% of bAbI+ with only 5 input dialogues from bAbI. It also shows that MemN2Ns can also generalise remarkably well. Although as discussed below, this result is misleading on its own as the MemN2Ns are very poor at generating the final API calls correctly on both the bAbI and bAbI+ data, and are thus making too many semantic mistakes.

### 3.5.2 Experiment 2: Semantic Accuracy

The results from Experiment 1 on their own can be misleading, as correct prediction of system responses does not in general tell us enough about how well the models are interpreting the dialogues, or whether they are doing this with a sufficient level of granularity. To assess this, in this second experiment, we measure the semantic accuracy of each model by looking *exclusively* at how accurately they predict the final API calls in the bAbI and bAbI+ datasets. For the MemN2N model, we follow the same overall procedure as in the previous experiment: train on bAbI data, and test on bAbI+.

#### 3.5.2.1 Results: Prediction of API Calls

**DyLan results.** Successful parsing of all the dialogues in the bAbI and bAbI+ datasets as shown above does not mean that the semantic representations compiled for the dialogues were in fact correct. To measure the semantic accuracy of the DS-TTR parser DyLan we programmatically checked that the correct slot values — those in the API call annotations — were in fact present in the semantic representations produced by the parser for each dialogue (see Fig. 2.23 for example semantic representations). We further checked that there is no other incorrect slot value present in these representations. The results showed that the parser has 100% semantic accuracy on both bAbI and bAbI+. This result is not surprising, given that DS-TTR is a general model of incremental language processing, including phenomena such as self-corrections and restarts (see Hough, 2014 for details of the model).

**MemN2N results — small data setup.** Given just 1 to 5 training instances from bAbI as in the previous experiment, the mean API call prediction accuracy of the MemN2N model is nearly 0 on both bAbI and bAbI+. This is not at all unexpected, since we see prediction of the API calls as an inherently *generation* process, unlike the prediction of system turns which can be done on a retrieval/look-up basis alone. For this, the model needs to observe the different word sequences

| Train / test set configuration | Train accuracy | Test accuracy |
|---|---|---|
| bAbI/ bAbI | 100 | 100 |
| bAbI/ bAbI+ | 100 | 28 |
| bAbI+/ bAbI | 67 | 99 |
| bAbI+/ bAbI+ | 72 | 53 |

TABLE 3.1: API call accuracy (%) of the MEMN2N trained on the full dataset

that might determine each parameter (slot) value, and observe them with sufficient frequency and variation. This is unlike a semantic parser like DS-TTR, that produces semantic representations for the dialogues as a result of the structural, linguistic knowledge that it embodies.

**MEMN2N results — full data setup.** Nevertheless, we were also interested in the general semantic robustness of the MEMN2N model to the transformations in bAbI+, i.e. how well does the MEMN2N model interpret bAbI+ dialogues, when trained on the full bAbI dataset? Does it then learn to generalise to (process) the bAbI+ dialogues with sufficient semantic accuracy?

Our hypotheses are that (i) given the positional encoding of memory vectors in the MEMN2N model and the underlying attention mechanism, it would be able to learn to process bAbI+ dialogues given that it was trained on similar data, resulting in an insignificant drop in performance from bAbI to bAbI+ data; (ii) a lot more data would be needed to learn to process the bAbI+ structures; and (iii) if trained on bAbI data, there would be a very significant drop in performance on bAbI+ with incorrect API calls predicted as a result of incorrect weightings and total lack of opportunity to learn the meaning of words such as "no" or "sorry" which trigger the self-corrections and restarts.

Table 3.1 shows that we can fully replicate the results reported in Bordes et al. (2017): the MEMN2N model can predict the API calls with 100% accuracy, when trained on the bAbI trainset and tested on the bAbI testset. But when this same model is tested on bAbI+, the accuracy drops majorly to 28%, making any dialogue system built using this model unusable in the face of more diverse dialogue data — thus confirming our hypothesis (iii). This is further discussed below.

#### 3.5.2.2 How Much Data Is Enough Data?

Given the results obtained so far, we are next interested in: (1) how robust MEMN2Ns are to the surface transformations in bAbI+ when trained on bAbI; (2) can MEMN2Ns learn to interpret bAbI+ when they are in fact trained on similar data that actually contain the bAbI+ structures — i.e. when trained on bAbI+; and (3) if so, how much bAbI+ data is needed for this. While (1) is a question about generalisation properties of a model, (2) & (3) are about potential in principle and/or practical limitations of MEMN2Ns to learn to interpret dialogues containing, e.g.

| Training bAbI+ dialogues | Memory hops | Embedding size | Train acc. | Test acc. |
|---|---|---|---|---|
| 2000 | 2 | 128 | 72.5 | 57.5 |
| 5000 | 2 | 128 | 72.7 | 60.7 |
| 10.000 | 2 | 128 | 72.8 | 65.8 |
| 50.000 | 1 | 128 | 82.6 | 78.2 |
| 100.000 | 1 | 64 | 83.3 | 80.5 |

TABLE 3.2: MEMN2N API call accuracy (%) with extended training data

self-corrections where utterances contain both the correct, and an incorrect (and subsequently repaired) slot value (e.g. "for four sorry five people"). To answer (1) we therefore train the model on the bAbI dataset and test on bAbI+; and to answer (2) & (3), we train the model on the bAbI+ train set and test it on the bAbI+ test set. Furthermore, in order to explore the impact of the amount of training data on the model's performance, we perform the latter experiment with varying train set size, as well as varying the hyperparameters: embedding size and the number of hops. The extended training data is obtained in the same way as the initial bAbI+ dataset: we go over the same original bAbI dialogues and keep randomly mixing in the incremental modifications.

Table 3.2 shows how MEMN2N performs on the same initial, fixed bAbI+ test set, when trained on progressively more data and up to 100,000 bAbI+ dialogues. As MEMN2N's performance on bigger data highly depends on the model's hyperparameters, in this experiment we perform a grid search over the number of memory hops (1, 2, 3), and the embeddings dimensionality (32, 64, 128) for each train set size independently — everything else is fixed as in the previous experiment. Only the best performing hyperparameter configuration for each of the train set sizes are included in the table.

The results confirm hypothesis (ii) above, i.e. that MEMN2Ns are in principle able to learn to process the incremental dialogue phenomena in bAbI+ but that they require tens of thousands of training instances for this: even with 100,000 dialogues, the semantic accuracy on the original test set stands at 80.5%.

## 3.6 Discussion

### 3.6.1 MEMN2N Analysis

The MEMN2N model was able to predict system responses remarkably well, even when trained on very few training instances. But results from Experiment 2 above showed that this was misleading: the MEMN2Ns were making a drastic number of semantic mistakes when interpreting the dialogues, both in the bAbI and bAbI+ datasets. Even when trained on the full bAbI dataset,

the model performed badly on bAbI+ in terms of semantic accuracy. We diagnose these results as follows:

**Problem complexity**. The first thing to notice is that in bAbI dialogue Task 1, the responses are highly predictable and stay constant regardless of the actual task details (slot values) up to the point of the final API calls; and further, that the prediction of API calls is a *generation* process, unlike the prediction of the system turns, which is retrieval-based. This, in our view, explains the very large difference in MemN2N performance across the two prediction tasks.

**Model robustness to the bAbI+ transformations**. The variations introduced in bAbI+ are repetitions of both content and non-content words, as well as of additional incorrect slot values. The model was working in the same setup as DyLan, therefore none of these variations could be treated as unknown tokens for either system. Although in the case of MemN2N, some of the mixed-in words never appeared in the training data, and bAbI+ utterances were augmented significantly with those words — so it was interesting to see how such untrained embeddings would affect the latent memory representations inside MemN2N. The resulting performance suggests that there was no significant impact on MemN2N from these variations as far as the predicting system responses was concerned. But the incorrect slot values introduced in self-corrections affect the system's task completion performance significantly, only appearing at the point of API call predictions.

It is worth noting that when trained on extended amounts on bAbI+ data with sufficient representation of target speech phenomena, MemN2N compensates for the most of the initial performance drop. However, it attains a reasonable level of accuracy ($> 80\%$) when trained on 100,000 bAbI+ dialogues which renders it potentially impractical for real-world tasks with more complex data distributions.

### 3.6.2 DyLan Analysis

The linguistically informed DyLan-based model we used in this chapter has the following conceptual advantages over previous approaches to dialogue system development:

— word-by-word incremental (and thus more natural) language understanding, dialogue management, and generation[4];

— a complete dialogue system for a new task can be automatically induced, using only 'raw' data — i.e. successful dialogue transcripts;

— wide-coverage, task-based dialogue systems can be built from much smaller amounts of data as shown in Section 3.5.

[4]Applicable to the full-fledged BABBLE setup as described in Section 2.5.2.

This final point bears further examination. Since it is an empirically adequate model of incremental language processing in dialogue, the DS-TTR grammar is required to capture interactional variants such as question-answer pairs, over- and under-answering, self- and other-corrections, clarification, split-utterances, and ellipsis more generally. As we showed in Section 3.5, even if most of these structures are not present in the seed example(s), the final system is able to handle them, thus resulting in a very significant generalisation around the original data. It can be said that the DyLan setup is a carefully tuned rule-based system, thus perhaps rendering these results trivial. But we note that the results here are not due to ad-hoc constructions of rules/lexicons, but due to the generality of the grammar model, and its attendant incremental, left-to-right properties; and that the same parser can be used in other domains. Furthermore, the ability to process self-corrections, restarts, etc. "comes for free", without the need to add or posit new machinery.

The generalisation results we report above for DyLan follow entirely from the knowledge present within the grammar as a computational model of dialogue processing and contextual update, rather than this having been learned from data. Applying the full RL method of BABBLE (Section 2.5.2) would have meant that the system would actually discover many interactional and syntactic variations that are not present in bAbI, nor in bAbI+.

## 3.7 Conclusions

In this chapter, we have evaluated the generalisation properties of a purely linguistically informed model based on the dialogue semantic parser DyLan, and compared it to a neural response retrieval model MemN2N. We did so by putting them in a controllable environment of bAbI Dialog Tasks and performing a series of experiments assessing their generalisation potential to more interactional variations in the input data (i.e., generalisation from seed examples to the full dataset), as well as the models' robustness to the simulated spoken language phenomena, i.e. self-corrections, hesitations, and restarts — represented in our bAbI+ corpus.

Our experiments show that MemN2N lacks the ability to generalise to such phenomena, and performs poorly when confronted with such variations even within synthetic, programmatically generated dialogue data. Our experiments further show that although this particular model is in principle able to learn to process disfluent dialogue phenomena, it requires an impractically large amount of data to do so. The results in this chapter therefore shed significant light on the currently ambiguous accuracy results reported for end-to-end systems (Bordes et al., 2017).

On the other hand, experiments with DyLan show that it can process 74% of the bAbI Dialog Task 1 even when only exposed to 0.13% of the data (5 dialogues); it can in addition process 65%

of bAbI+. That is in contrast to MemN2N which was not robust to the structures we introduced in bAbI+, even when trained on the full bAbI dataset.

The above results give us the following insights on the two possible ways towards attaining a practical level of data efficiency in dialogue. Firstly, an inductive bias in the form of a linguistically informed model can be an efficient approach in an extreme case of 1-shot/few-shot generalisation. However, the bottleneck here is the linguistic resource itself, the incremental dialogue grammar of DyLan in our case. Parsing natural speech is generally challenging, especially in case of freeform conversation, so producing wide-coverage grammars for this purpose can be a notoriously hard task. The second path featuring neural models has a definite advantage here from the Natural Language perspective: given a sufficient amount of data to represent the phenomena of interest, models like MemN2N are able to adjust and process it adequately from the target task's perspective. Scaling to an unknown domain or a foreign language for that matter will require nothing more than a sufficiently representative dataset. The challenge here though is how much data is considered sufficient for that, and our controlled experiments with MemN2N show that its actual data requirements are quite far from practical. Therefore, the vital step towards real-world applicability of such systems is to reduce the amount of required training data and annotations closer to what can be considered practically data-efficient without the loss of the models' generalisation potential.

In the next chapter, we are going to explore the purely data-driven approach to dialogue system bootstrapping. That is, we will focus on learning transferable representations of dialogue capturing lexical and interactional similarity (as discussed in Section 2.4) and their applicability for reducing the data consumption of goal-oriented dialogue systems.

# Chapter 4

# Learning Transferable Dialogue Representations

As we saw in the previous chapter, both linguistically informed and purely data-driven methods have their potential in terms of data efficiency. The principal advantage of machine learning methods is that they generally do not require constructing and maintaining syntactic/semantic grammars which are very challenging to build, especially in the setting of spontaneous spoken language. The problem with machine learning methods though is, as we also saw previously, their performance depends to a high degree on the training data, and the overall data consumption of those models significantly reduces their flexibility in practical setups. A possible solution to that is to use large sources of domain-independent data in order to obtain the common language and dialogue representation and then transfer it to the specific problem domain, with minimal fine-tuning to the available in-domain data. As outlined in Chapter 2, transfer learning has already proved to be a very efficient technique in computer vision and is being actively adopted in Natural Language Processing — in this chapter, we are going to explore its applicability to dialogue.

Specifically, we are going to present the Dialogue Knowledge Transfer Network (or DiKTNet), a goal-oriented dialogue response generation model designed for few-shot learning, i.e. training only using a small number of complete in-domain dialogues. The key underlying concept of this model is transfer learning: DiKTNet makes use of the latent text representation learned from several sources ranging from large-scale general-purpose textual corpora to similar dialogues in the domains different to the target one. We use the evaluation framework of Zhao and Eskénazi (2018) and mainly compare our approach to theirs — and, similarly to them, we use the Stanford Multi-Domain dialogues dataset (Eric et al., 2017). While Zhao and Eskénazi's method does not require complete in-domain dialogues and uses annotated utterances instead (and is therefore presented as "zero-shot"), we show that our model achieves superior performance with roughly

| | |
|---|---|
| **Driver:** | I need to find out the time and parties attending my optometrist appointment |
| **Car:** | I have 3 appointments scheduled, with Alex, your sister, and Jeff. Which are you referring to? |
| **Driver:** | I want to know about the one that Alex is joining me at |
| **Car:** | That optometrist appointment is at 4 pm. |
| **Driver:** | Thanks |
| **Car:** | no problem |

(a)

| Event | Time | Date | Party | Agenda |
|---|---|---|---|---|
| opt. appt | 10 am | Thursday | sister | — |
| dinner | 8 pm | the 13th | Ana | — |
| opt. appt | 7 pm | the 20th | Jeff | — |
| opt. appt | 4 pm | the 13th | Alex | — |
| ... | ... | ... | ... | ... |

(b)

TABLE 4.1: Example dialogue from SMD (a) with the corresponding knowledge base snippet (b) — driver is the user, car is the system

the same amount of data (with respect to in-domain utterances) while requiring no annotations whatsoever.

## 4.1 Few-Shot Dialogue Generation

We first describe the task we are addressing in this chapter, and the corresponding base model. Specifically, here we work with dialogue data that is organised into multiple non-overlapping domains. We say that *source domains* can be used to train the model, whereas the *target domain* is mainly used for evaluation. Different data-efficient setups assume the corresponding usage of target-domain data. As such, in zero-shot learning, the full target dialogues are not used at all; few-shot learning assumes fine-tuning to several target dialogues and then evaluating on the rest — we will be working in the latter setting.

Our domains here are in-car navigation, weather information, and appointment scheduling — as represented in the SMD dataset we are going to work with (see Section 2.7 for a short description). In a single experiment, we work with 2 of those domains as source, and the third one becomes the target domain. An example dialogue from SMD is shown in Table 4.1 — every dialogue in the dataset comes with a snippet from the underlying Knowledge Base working a as simulation of the domain-specific search API. The dialogue system's task then is to (1) train on the available data from the source domains, while learning to extract relevant data from the KB snippets along the way, (2) fine-tune to the given seed dialogues + KB snippets in the target domain, and (3) predict responses for the rest of the target-domain dialogues.

## 4.2 The Base Model

In our approach, we are building on top of HRED — in particular, the copy-augmented model (Merity et al., 2017). HRED was discussed earlier in Section 2.3. Here, we are providing its

Negative log-likelihood (NLL) optimisation objective which we are going to be building upon:

$$\mathcal{L}_{\text{HRED}} = \log p_{\mathcal{F}^d}(y_{\text{sys}} \mid \mathcal{F}^e(c, x_{\text{usr}})) \tag{4.1}$$

where $x_{\text{usr}}$ is the user's query, $y_{\text{sys}}$ is the system's response, $c$ is the dialogue context, and $\mathcal{F}^e$ and $\mathcal{F}^d$ are respectively the hierarchical encoder and the decoder.

We work with goal-oriented dialogues where part of the system's task is to provide the integration with an underlying database or API via the KB snippets, as described above. Given that such information may contain unseen token sequences for the most part, especially in the target domain, we use a copy mechanism in order to be able to directly transfer those tokens from the input into the system's responses. More specifically, we represent the KB info as token sequences and concatenate them to the dialogue context similarly to the CopyNet setup of Eric et al. (2017), the only difference is the actual copy mechanism implementation for which we employ the Pointer-Sentinel Mixture model (Merity et al., 2017; Zhao and Eskénazi, 2018) — see the description of these models in Sections 2.3.5. Our model is thus supposed to produce goal-oriented responses for target dialogue contexts using the soft decision — whether to generate the next word from the vocabulary or to copy a token from earlier in the dialogue history (or at some position in the knowledge base response).

## 4.3 Dialogue Knowledge Transfer Networks

Transfer learning is considered the key means for efficient training with minimal data, and our DiKTNet model essentially introduces several knowledge-transfer augmentations to the base HRED model described above. DiKTNet training is performed in two stages described below.

### 4.3.1 Stage 1. Dialogue Representation Pre-training

Dialogue structure — e.g. word sequences — is highly specific to a given domain or task, and the meaning of conversational utterances is highly contextual, i.e. similar utterances may have different meanings depending on the context. Nevertheless, there is a lot of similarity in dialogue structure — i.e. sequences of dialogue actions — across domains, e.g. a conversation normally starts with a mutual greeting and a question is very often followed by an answer. Here, we propose to exploit this phenomenon in the form of learning a latent dialogue action representation in order to better capture the dialogue structure by abstracting away from surface forms. Crucially, we learn such representation from MetaLWOz (derived from *Meta-Learning Wizard-of-Oz*, Lee

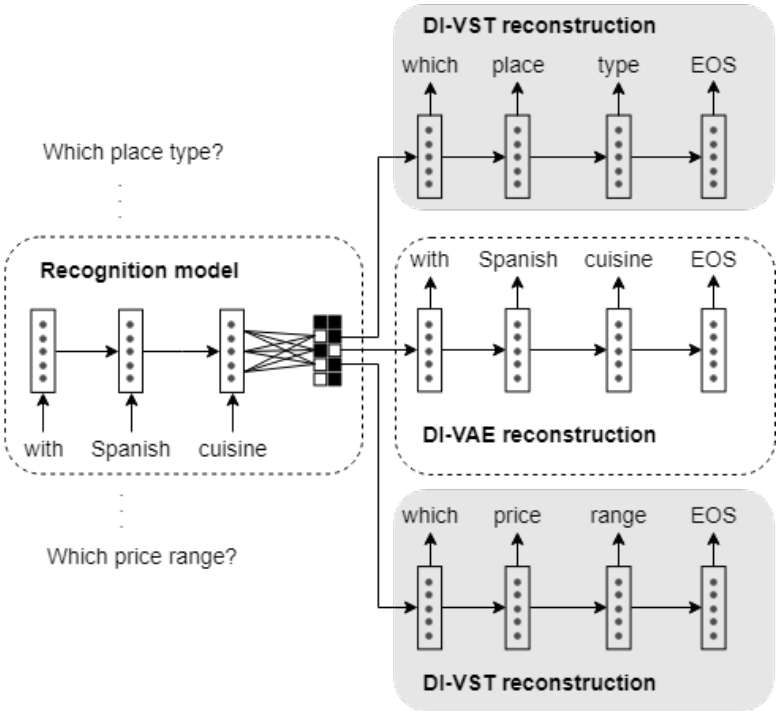



FIGURE 4.1: DI-VAE and DI-VST (DIKTNET Stage 1)

et al., 2019a), a dataset specifically created for the purposes of meta-learning and transfer learning and consisting of human-human conversations in 51 unique domains — see the description of the dataset in Section 2.7 and some statistics below in Section 4.5.

For this stage of training we use unsupervised, variational autoencoder-based (VAE) representation learning, and we will be particularly focusing here on a specific framework of discrete information (DI) sentence representations of Zhao et al. (2018). As will be discussed later in Section 4.7.2, the underlying variational autoencoding techniques facilitate more stable training and result in more efficient representation models as compared to the more widely-used conventional VAE. As was introduced in Section 2.3.2 DI-VAE model differs from a classic, continuous VAE in that it (1) implicitly promotes mutual information between the model's input and its latent code and (2) uses Batch Prior Regularisation technique for the calculation of the KL divergence in the case of discrete latent codes. DI-VAE and its skip-thought counterpart DI-VST are visualised in Figure 4.1.

In the downstream DIKTNET model, we use DI-VAE autoencoder in order to obtain the representation of the user's query: $z_{\text{usr}}$ = DI-VAE($x_{\text{usr}}$). DI-VST, in turn, is used to obtain a prediction of the system's action $z_{\text{sys}}$ in the discretised latent form given the user's input $x_{\text{usr}}$ as well as the full dialogue context $c$. For that, DI-VST autoencoder is used as part of a hierarchical, context-aware encoder-decoder response generation model — following Zhao et al., 2018, we refer to it as Latent Action Encoder-Decoder, or LAED (discussed in Section 2.3.3), its optimisation objective is as follows:

$$\mathcal{L}_{LAED} = \mathbb{E}_{q_{\mathcal{R}}(z_{\text{sys}}|y_{\text{sys}})p(y_{\text{sys}},c,x_{\text{usr}})}\Big[\log p_{\pi}(z_{\text{sys}} \mid c, x_{\text{usr}}) + \log p_{\mathcal{F}}(y_{\text{sys}} \mid z_{\text{sys}}, c, x_{\text{usr}})\Big] \tag{4.2}$$

where $\mathcal{F}$ is the decoder generating the system's response $y_{\text{sys}}$, and $\pi$ is the 'policy' feed-forward network predicting $z_{\text{sys}}$ from the dialogue context $c$ and the user's last turn $x_{\text{usr}}$. As noted in out introduction of LAED (Section 2.3.3), the recognition model $q_{\mathcal{R}}(z_{\text{sys}} \mid y_{\text{sys}})$ conditioned on the gold system's response is only used during training and is discarded at prediction time.

Our intuition behind using different models for representing the user's query and the system's latent action follows empirical results of Zhao et al. (2018) who showed that DI-VAE is better at capturing specific words of an utterance, while DI-VST represents the overall dialogue action better. We train these two models on MetaLWOz in an unsupervised way with the objectives as described above, and use their discretised latent codes $z_{\text{usr}}$ and $z_{\text{sys}}$ respectively in the downstream model at the next stage of training.

### 4.3.2 Stage 2. Transfer

At this stage, we train directly for our target task, few-shot dialogue generation, and thus go back to the model described in Section 4.1. While the training procedure of this model naturally assumes *domain transfer*, we will provide it with more sources of textual and dialogue knowledge of varying generality described below.

As opposed to direct domain transfer, we incorporate domain-general dialogue understanding from the LAED representation trained on MetaLWOz at the previous stage. LAED captures the background top-down dialogue structure: sequences of dialogue acts in a cooperative conversation, latent dialogue act-induced clustering of utterances, and the overall phrase structure of spoken utterances. We incorporate this information into the model by conditioning HRED's decoder on the combined latent codes from Stage 1 and refer to this model as HRED +Stage1. Its optimisation objective is as follows:

$$\mathcal{L}_{\text{HRED+Stage1}} = \mathbb{E}_{p(x_{\text{usr}},c)p(z_{\text{usr}}|x_{\text{usr}})p_{\pi}(z_{\text{sys}}|x_{\text{usr}},c)}\Big[\log p_{\mathcal{F}^d}\Big(y_{\text{sys}} \mid \big\{\mathcal{F}^e(x_{\text{usr}}, c), z_{\text{usr}}, z_{\text{sys}}\big\}\Big)\Big] \tag{4.3}$$

where $z_{\text{usr}}$ and $z_{\text{sys}}$ are respectively samples obtained from the DI-VAE user utterance model and the LAED system action model, and $\{\cdot\}$ is the concatenation operator.

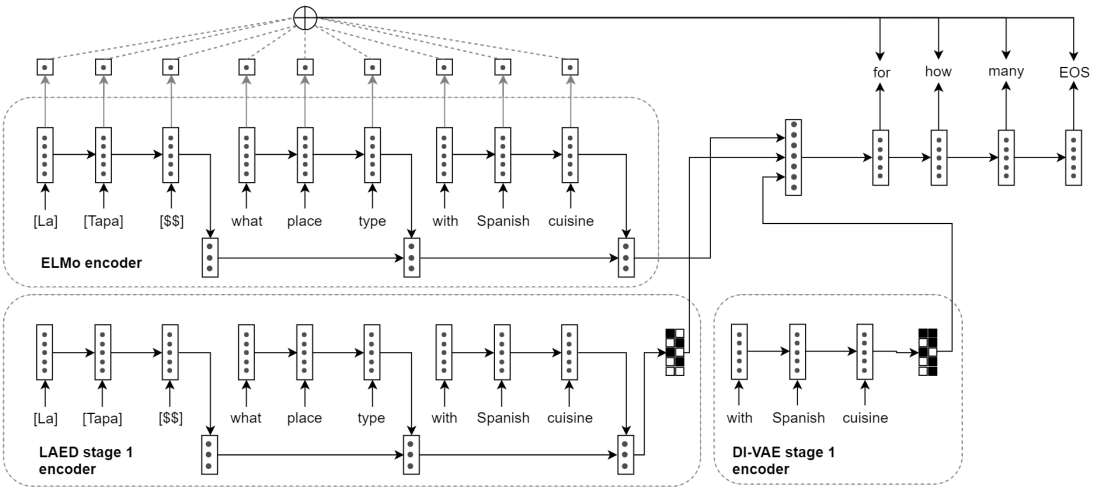



FIGURE 4.2: DIKTNET Stage 2 (tokens in brackets are KB data)

The last, most general source of knowledge we use is a pre-trained ELMo model (Peters et al., 2018). Apart from using an underlying bidirectional RNN encoder, ELMo captures both token-level and character-level information which is especially crucial in understanding unseen tokens and KB items in the underrepresented target domain. The HRED model with ELMo as the utterance-level encoder is referred to as HRED +ELMo.

Finally, DIKTNET is the HRED augmented with both ELMo encoder and sentence representations (DI-VAE and LAED) from Stage 1.

DIKTNET is visualised in Figure 4.2. The model (as well as its variants listed above) is implemented in PyTorch (Paszke et al., 2017), and the code is openly available[1].

## 4.4 Baselines

We perform an exhaustive ablation study of DIKTNET by comparing it to all of its variations mentioned above: HRED, HRED +ELMo, and HRED +Stage1. In addition to that, we have the HRED +VAE — a HRED +Stage1 counterpart for which we use a regular, continuous VAE behind DI-VAE and DI-VST in order to determine the impact of discretised latent codes (see Eq. 2.14 for the corresponding objective function).

Furthermore, we compare DIKTNET to the previous state-of-the-art approach, Zero-Shot Dialogue Generation (Zhao and Eskénazi, 2018). This model did not use any complete in-domain dialogues but instead it relied on annotated utterances in all of the domains. We use it as-is (ZSDG), as well its variation as follows.

[1] http://tiny.cc/diktnet

We make use of ZSDG's central idea of using NLU-annotated in-domain utterances as 'domain descriptions' that facilitate bridging dialogue understanding across domains, but instead of using manually annotated utterances, we employ automatic NLU markup. Our annotations include:

— Named Entity Recognition — Stanford NER model ensemble of case-sensitive and case-less models (Finkel et al., 2005),

— Date/time markup — Stanford SUTime (Chang and Manning, 2012),

— Wikidata entity linking — Yahoo FEL (Blanco et al., 2015; Pappu et al., 2017).

We serialise annotations from these sources into token sequences and make domain description tuples out of all the utterances in the source and target domains. In this way, most of our domain descriptions share the structure and content of the original ones.

For example, for the phrase *'Will it be cloudy in Los Angeles on Thursday?'*:

— the original ZSDG annotation is `"request #goal cloudy #location Los Angeles #date Thursday"`,

— our NLU annotation is `"LOCATION Los Angeles DATE Thursday"`.

We have two models in this setup, with (NLU_ZSDG+Stage1) and without the use of Stage 1 representations (NLU_ZSDG) respectively.

## 4.5 Datasets

For the latent representation learning, we use MetaLWOz described in Section 2.7. All the domains available in the MetaLWOz dataset are listed in Table A.2 of Appendix A, and some example dialogues can be found there in Section A.

Our target dataset is the Stanford Multi-Domain (SMD) dialogues corpus (Eric et al., 2017) described in Section 2.7. SMD contains human-human goal-oriented dialogues in three domains, with a simulation of the underlying search API: each dialogue in SMD comes with a knowledge base snippet representing the result of implicitly querying the API, with the KB schema (i.e. columns names and the data types) being specific to each domain. Although sharing some common features (the setting of an intelligent in-car assistant and the use of the underlying KB), the dialogues differ significantly across domains which makes the domain transfer sufficiently challenging. The statistics of the are shown in Tables 4.3 and 4.2, respectively.

| | |
|---|---|
| Domains | 51 |
| Dialogues | 40,388 |
| Avg. dialogue length | 11.91 |

TABLE 4.2: METALWOZ dataset statistics

| | Navigation | Weather | Schedule |
|---|---|---|---|
| Dialogues | 800 | 797 | 828 |
| Utterances | 5248 | 4,314 | 3,170 |
| Avg. dialogue length | 6.56 | 5.41 | 3.83 |

TABLE 4.3: Stanford multi-domain dataset statistics (trainset)

In our experiments, we make sure that all the source data have no domain overlap with the target dialogues we're evaluating on, therefore we make our training setup dynamic by excluding the specific METALWOZ domains based on the target SMD one, such that:

— for the Navigation target domain in SMD, we exclude METALWOZ's Store Details domain,

— for Weather, we exclude Weather Check,

— for Schedule, we exclude Update Calendar and Appointment Reminder.

## 4.6 Experimental Setup and Evaluation

Our few-shot setup is as follows. Given the target domain, we first train Stage-1 model(s) on the METALWOZ data, having filtered source domains as described above. We used a DI-VAE and a DI-VST-based LAED, both of the size $10 \times 5$.

Next, having trained and Stage-1 models, we train DIKTNET on all the source domains from the SMD dataset without further fine-tuning of DI-VAE/LAED. We also fine-tune to a portion of target-domain data (thus working in a few-shot setup) by sampling the target dialogues together with their KB info, varying the amount of those from 1% to 10% of all the available target data.

For the NLU_ZSDG setup, we annotated all available SMD data and randomly selected a subset of 1000 utterances from each source domain, and 200 utterances from the target domain. For source domains, this number amounts to roughly a quarter of all available training data — we chose it in order to make use of as much annotated data as possible while keeping the domain description task secondary. We made sure to keep under roughly the same target-domain data requirements as the ZSDG baseline.

For evaluation, we follow the approach of Zhao and Eskénazi (2018) and report BLEU and Entity F1 scores. Given the non-deterministic nature of our training setup, we report means and variances of our results over 10 runs with different random seeds.

We also perform an additional evaluation of DIKTNET's performance with extended amounts of target data and compare it to the original results for the SMD dataset reported by Eric et al.

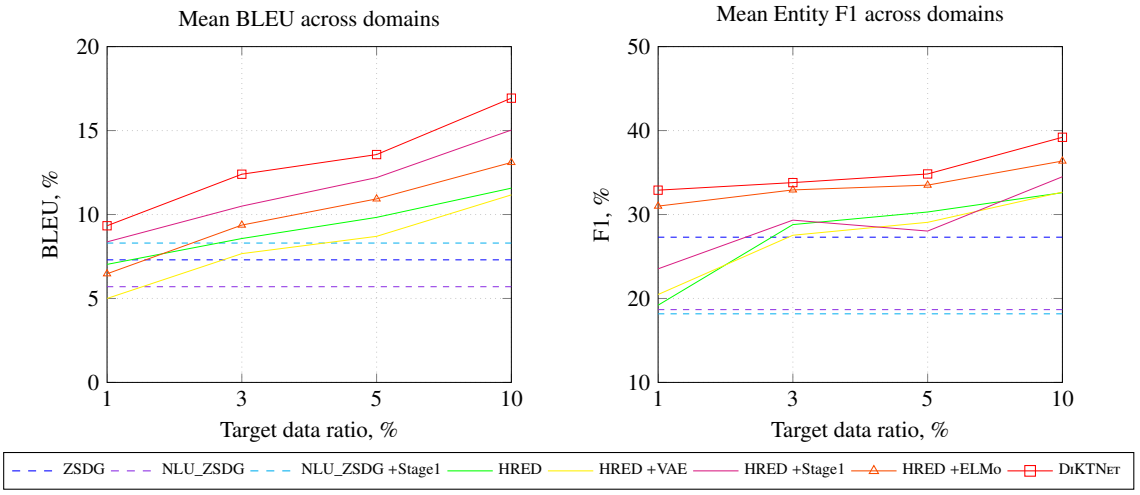



FIGURE 4.3: Models' performance on the SMD dataset. Dashed are ZSDG baselines that use a fixed set of annotated in-domain utterances instead of raw dialogues (see Section 4.4)

(2017) upon the dataset introduction. Their model — Key-Value Retrieval Network (KVRET), a variant of copy-augmented SEQ2SEQ with a separate copying mechanism for the KB snippets — was trained with all the available data for a given domain. In this evaluation, we average BLEU scores across all 3 SMD domains in order to be consistent with the form that the corresponding results are presented in the original paper.

We train our models with the Adam optimiser (Kingma and Ba, 2015) with the learning rate of 0.001. Our hierarchical models' utterance encoder is an LSTM cell (Hochreiter and Schmidhuber, 1997) of the size 256, and the dialogue-level encoder is a GRU (Cho et al., 2014b) of the size 512.

## 4.7 Results and Discussion

Our results are shown in Figure 4.3 — see also Table A.1 of the Appendix A for a more detailed breakdown. The former contains BLEU and Entity F1 scores averaged over target domains, and the latter has the corresponding values for each domain separately, showing means and variances. Our objective here is maximum accuracy with minimum training data.

### 4.7.1 Results for the Few-Shot Setup

It can be seen that few-shot models with DI-VAE/LAED representation are the best performing models for this objective. While improvements upon ZSDG can already be seen with simple HRED in a few-shot setup, the use of the Stage-1 representation and domain-general ELMo encoding helps significantly reduce the amount of in-domain training data needed: at 1% of in-domain dialogues, we see that DIKTNET consistently and significantly improves upon ZSDG in

| |
|---|
| Where can I go shopping? |
| Where does my friend live? |
| Where can I get Chinese food? |
| Where can I go to eat? |
| Can you please take me to a coffee house? |
| I'd like to set a reminder for my meeting at 2pm later this month please. |
| What is the time and agenda for my meeting, and who is attending? |
| Schedule a lab appointment with my aunt for the 7th at 1pm. |
| Schedule a calendar reminder for yoga with Jeff at 6pm on the 5th. |
| Car I'm desiring to do some shopping: which one is it the nearest shopping ... |
| ... center? Anything within 4 miles? |
| Get the address to my friend's house that i could get to the fastest |
| Car I need to get to my friends house, it should be within 4 miles from here |

TABLE 4.4: Selected clusters of utterances sharing the same DI-VAE codes

every domain. In SMD, with its average dialogue length of 5.25 turns, 1% of training dialogues amounts to approximately 40 in-domain training utterances. In contrast, the ZSDG setup used approximately 150 training utterance-annotation pairs for each domain, including the target one, totalling about 450 *annotated* utterances.

Although in our few-shot approach we use full in-domain dialogues, we end up having significantly less in-domain training data, with the crucial difference that none of those has to be annotated for our approach. Therefore, the method we introduced improves upon the previous best approach in both accuracy and data-efficiency.

In turn, the results of the ZSDG_NLU setup demonstrate that single utterance annotations, if not domain-specific and produced by human experts, do not provide as much signal as full dialogues, even without annotations at all. Even the significant number of such annotated utterances per domain did not make a difference in this case.

We would also like to point out that, as can be seen in the table, our results have quite high variance — the main source of it is the nature of our training/evaluation setup where we average over 10 runs with 10 different sets of seed dialogues. However, in the majority of cases with comparable means, DIKTNET has a lower variance than the alternative models at the same percentage of seed data. And in the extreme case with 1% target data, DIKTNET improves on all the other models in terms of both means and variances.

### 4.7.2 Discussion of the Latent Representations

The comparison of the setups with different latent representations also gives us some insight: while the VAE-powered HRED model improves on the baseline in multiple cases, it lacks generalisation potential compared to the DI-VAE/LAED setup. The reason for that might be the inherently more stable training of DI-VAE/LAED due to their modified objective function,

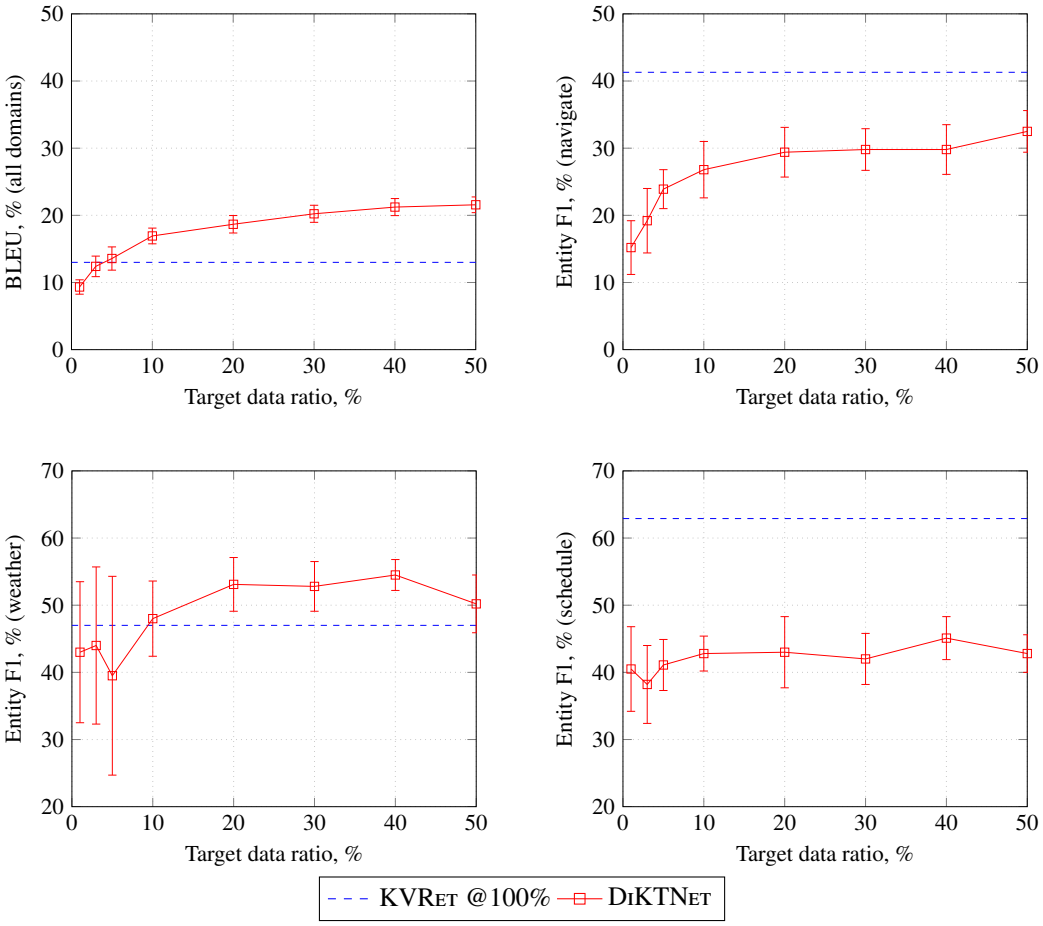



FIGURE 4.4: DIKTNET performance with extended amounts of target data used for training

which in turn results in a more informative representation providing better generalisation. For instance, with the 'vanilla' VAE setup, we immediately experienced the commonly reported vanishing KL term problem (discussed previously in Section 2.3.3) which effectively turned our VAE into a significantly overfitted AE. With the discrete-information models — both DI-VAE and DI-VST — we did not experience this problem even without using any techniques that are considered crucial in the training of VAEs (Bowman et al., 2016).

In order to have a glimpse into the DI-VAE-produced clustering, in Table 4.4 we present a snippet of the utterance clusters sharing the same, most frequent latent codes throughout the dataset (the clustering is obtained with DI-VAE model trained on every domain but 'Store details', i.e. the one for the evaluation on 'Navigate' SMD domain). From this snippet, it can be seen that those clusters work well for domain separation, as well as capturing dialogue intents.

### 4.7.3 Results with Extended Data

We performed an additional experiment with extended amounts of target data (see Figure 4.4). It showed that DIKTNET, when trained with as little as 5% of target data, can outperform a KVRET

| Domain | | Context | Gold response | Predicted response |
|---|---|---|---|---|
| schedule | <usr> | Remind me to take my pills | Ok setting your medicine appointment for 7pm | Okay, setting *a reminder to take your pills at 7 pm.* |
| | <sys> | What time do you need to take your pills? | | |
| | <usr> | I need to take my pills at 7 pm. | | |
| navigate | <usr> | Find the address to a hospital | Have a good day | *No problem.* |
| | <sys> | Stanford Express Care is at 214 El Camino Real. | | |
| | <usr> | Thank you. | | |
| weather | <usr> | What is the weather forecast for the weekend? | For what city would you like to know that? | For what city would you like *the weekend forecast for?* |

TABLE 4.5: DIKTNET's selected responses

trained using the entire dataset. Furthermore, with 50% of the target data, DIKTNET becomes more than twice as good as KVRet in terms of overall language generation.

However, goal-oriented metrics such as Entity F1 are more challenging to bootstrap. As such, DIKTNET outperforms KVRET on 'Weather' domain starting at 10% of the target data, but only has a trend on narrowing down the performance gap with KVRet on 'Navigate', and certainly needs more training data in the 'Schedule' domain.

The explanation for that might be that most of the dialogue entities come from the KB snippets which are the least represented resource in our setup. They are not available in MetaLWOz, and in SMD, KB snippets share little in common across domains. Therefore, in order to increase Entity F1, KB information should be directly copied to the output more efficiently — and increasing the robustness of the copy-augmented decoder is one of our future research directions.

### 4.7.4 Discussion of the Evaluation Metrics

We use BLEU as one of the main evaluation metrics in this work — we do it in order to fully conform with the setup of Zhao and Eskénazi (2018) which we base our work on. But while being widely adopted as a general-purpose language generation metric, BLEU might not be sufficient in the dialogue setting (see Novikova et al. (2017) for a review). Specifically, we have observed several cases where the model would produce an overall grammatical response with the correct dialogue intent (e.g. "You are welcome! Anything else?"), but BLEU would output a lower score for it due to word mismatch (e.g. "You're welcome!"; see more examples in Table 4.5). This is a general issue in dialogue model evaluation since the variability of possible responses equivalent in meaning is very high in dialogue. We think that putting more emphasis on the meaning of utterances, for example by incorporating external dialogue act tagging resources in the evaluation setup which, together with general language generation metrics like perplexity, can make for more robust evaluation criteria than word overlap.

## 4.8 Conclusion

In this chapter, we have introduced DIKTNET, a model achieving strong dialogue response generation performance in a few-shot setup, without using any annotated data. By transferring latent dialogue knowledge from multiple sources of varying generality, we obtained a model with superior generalisation to underrepresented domains. Specifically, we showed that our few-shot approach improves upon the previous best model on the Stanford Multi-Domain dataset while being more data-efficient, by requiring significantly less data none of which has to be annotated.

In the next chapter, we will continue our study of low-resource dialogue generation by addressing the problem of fast adaptation of a dialogue system to a new domain, at a greater scale of 47 information-seeking domains (i.e. a version of MetaLWOz which becomes our target dataset). As such, we will explore alternative ways of using the support in-domain data other than fine-tuning the base model on it.

# Chapter 5

# Dialogue Domain Adaptation

In this chapter, we take our research on few-shot dialogue modelling further and continue with the problem of fast domain adaptation for dialogue systems. As argued in the previous chapters, domain adaptation is the key approach to the development of data-efficient dialogue systems in the machine learning framework; here we are going to explore this problem at a greater scale, i.e. through the Eighth Dialog System Technology Challenge (DSTC), Fast Domain Adaptation task. Specifically, we propose the hybrid *Generative-Retrieval*[1] *Transformer, GRTr*[2] — a model leveraging knowledge transfer from a large-scale pre-trained general-purpose language model and combining it with the response retrieval logic. The model is able to maintain goal-oriented dialogue in a closed domain having only been exposed to a small set of in-domain dialogues as the domain description. Our hybrid model is ranked 1st on the MetaLWOz dataset as per human evaluation, and also performs competitively on automated metrics when compared to other baselines —both generation-only and retrieval-only models.

## 5.1 Fast Domain Adaptation of a Dialogue System

As we saw in the previous chapter, few-shot knowledge transfer is a promising way to adapt a goal-oriented dialogue system to a new domain. In this chapter, we are going to continue working in the framework of MetaLWOz dataset, but will increase the scale by making it our target dataset. Since MetaLWOz does not contain any goal-oriented annotations but overall represents cooperative information-seeking dialogue between two humans, this task focuses on predicting the utterances on the user's side. This can be considered a more challenging task since normally, user's utterances are less predictable than those of the system, and a prospective successful dialogue model should have a representation of the underlying user's goal in order to

[1] Following Radford et al. (2018) and Zhang et al. (2020), we use the term "generative" in the sense of predicting (or generating) the next word given the context, i.e. language generation.

[2] Code is available at http://tiny.cc/grtr

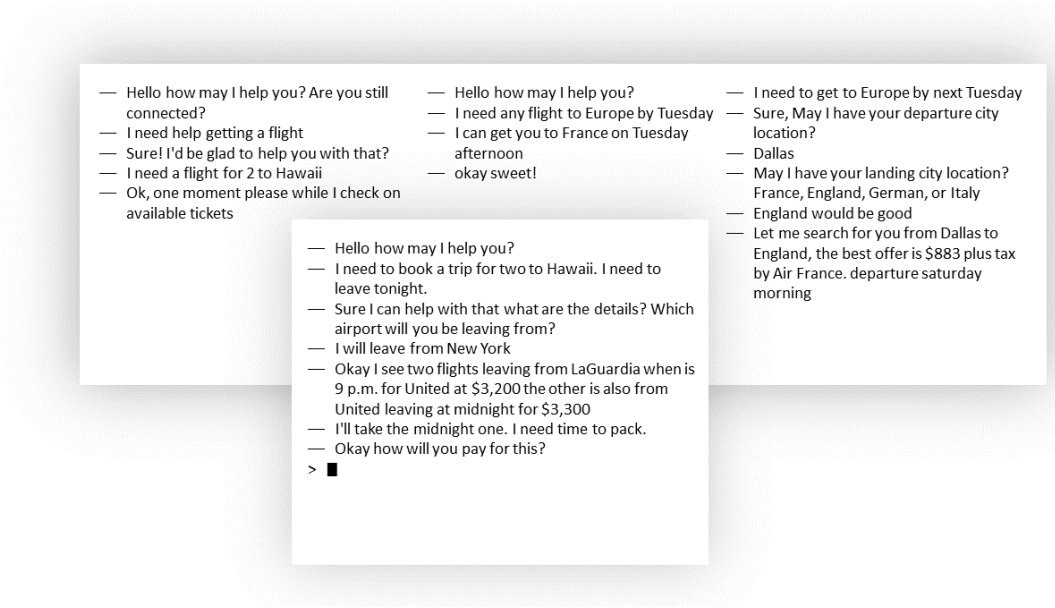



FIGURE 5.1: Example DSTC-8 support set (background) and target dialogue (foreground) in the travel domain

generate relevant queries to the system. Our task takes place within the DSTC-8 which we are going to describe next.

### 5.1.1 DSTC-8, Fast Domain Adaptation Task

In the Eight Edition of DSTC, its Domain Adaptation task focuses on building a model that predicts user responses for a goal-oriented dialogue system for which only limited in-domain data is available. The possible applications of an adaptive user-side dialogue model include Reinforcement Learning-based setups which are highly dependent on the quality of the user simulator, as well as data augmentation approaches for improving robustness and coverage of the target models.

The in-domain adaptation data could be collected from e.g. customer service transcripts, or written by the developers themselves. From this in-domain data, the *support set*, one would like to extrapolate responses to novel dialogue contexts (the *target*) — see example in Figure 5.1. However, the support set is typically too small to train a dialogue response generation model. Instead, the approach assumed in the challenge is to adapt (or fine-tune) a generic dialogue model trained on a large corpus of conversations over multiple *source* domains.

Technically, the problem setup is as follows: having trained the base model on the source domains, the model is then fed with one target dialogue and a support set at a time. The model's task is to predict the next user turn of the target dialogue, taking into account the support set before producing a prediction. At prediction time, each target dialogue is processed in isolation

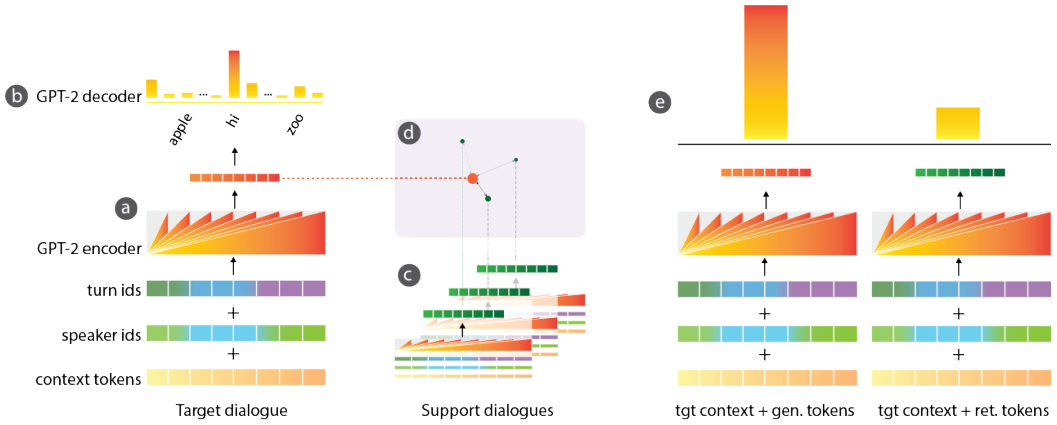



FIGURE 5.2: GRTR model diagram. We (a) encode the target dialogue context and (b) produce the 'generated candidate'; next, we (c) encode support dialogue contexts in a similar way, then (d) find the nearest 'support' neighbour and select its response as the 'retrieved candidate'; finally, we (e) rank the two candidates given the target context and produce the final result.

from other target dialogues, such that the model cannot use knowledge or state obtained from other target/support data.

## 5.2 Proposed Model

We use a language model pre-trained on a very large and diverse collection of textual data providing a strong language prior and then adapt the model for our tasks in the form of fine-tuning. Our base model is GPT-2 (Wolf et al., 2019), a transformer-based language model. In order to adapt GPT-2 for dialogue generation, we first augment the input embedding for each token in the dialogue with (1) a speaker tag embedding identifying the speaker and (2) a turn embedding, identifying the turn number in the current dialogue. These additional embedding matrices are learned solely using the dialogue data. The input token embeddings are then obtained by summing up these representations. We also add two task-specific output layers (or "heads") for our purposes: a language modelling (LM) head and a next-sentence prediction (NSP) classification head, both trained from randomly initialised parameters.

We fine-tune GPT-2 for response generation by minimising the negative log-likelihood of response tokens given the concatenation of dialogue context and the previous tokens in the response,

$$\begin{aligned} \mathcal{L}_{\text{LM}} &= -\log P_{\text{LM}}(X \mid C) \\ &= -\sum_{i=1}^{|X|} \log P_{\text{LM}}(x_i \mid x_{i-1}, ..., x_1, C), \end{aligned} \tag{5.1}$$

where $X$ is the response and $C$ is the dialogue context, i.e. the concatenation of the tokens in the previous utterances.

To predict the next sentence, we proceed as follows: given a context/response pair $(C, X)$, the classification head is trained to produce a binary label $y$, which is 1 if $X$ is the correct response given the context $C$, and 0 if $X$ is a distractor (a random utterance from the corpus). We minimise the following binary cross-entropy:

$$\mathcal{L}_{\text{NSP}} = -y \log P_{\text{NSP}}(y \mid X, C) - (1 - y) \log P_{\text{NSP}}(1 - y \mid X, C), \tag{5.2}$$

$$P_{\text{NSP}}(y \mid X, C) = \text{softmax}(f_{\text{NSP}}(h_{X,C})), \tag{5.3}$$

where $h_{X,C}$ is the last hidden state of the last GPT-2 layer after having encoded the concatenation of $X$ and $C$ and $f_{\text{NSP}}$ is the next-sentence prediction head (in our case a simple linear transformation). In practice, for each $(C, X)$ pair in the corpus, we sample 1 distractor $\bar{X}$.

We obtain a suitable dialogue prior by fine-tuning the modified GPT-2 model on the source domains with both the language modelling and next-sentence prediction tasks as described above, therefore minimising $\mathcal{L} = \mathcal{L}_{\text{NSP}} + \mathcal{L}_{\text{LM}}$.

### 5.2.1 Fine-tuning on Target Domains and Prediction

As every test dialogue in the target domain/task is accompanied with a small support set of dialogues from the same domain/task, we make use of this data by further fine-tuning the dialogue model on the support dialogues. Crucially, we make sure not to accumulate any information between test dialogues: after each fine-tuning on the support set, we reset the weights of the model to the dialogue prior obtained by the fine-tuning stage described in the previous section.

In order to add diversity to the responses, GPT-2 uses nucleus (top-$p$) sampling (Holtzman et al., 2020) during generation, i.e. the model's vocabulary $V$ is pruned to $V^p$, the smallest set such that

$$\sum_{x \in V^p} p(x \mid x_{1:i-1}, C) \geq p, \tag{5.4}$$

and the final distribution from which the words are sampled is rescaled as follows:

$$P'(x \mid x_{1:i-1}) = \begin{cases} \frac{P(x|x_{1:i-1},C)}{\sum_{x \in V^p} P(x|x_{1:i-1},C)} & \text{if } x \in V^{(p)} \\ 0, & \text{otherwise.} \end{cases} \tag{5.5}$$

### 5.2.2 Hybrid Generative-Retrieval Prediction

**Algorithm 1:** Hybrid generative-retrieval response prediction

**Input:** Enc — GPT-2 encoder

**Input:** Dec — GPT-2 decoder (language modelling head)

**Input:** NSP — GPT-2 next sentence prediction head

**Input:** $t$ — turn number to predict

**Input:** $X_tgt$ — target dialogue context of length $t-1$

**Input:** $X_sup$ — support dialogues (sequences of turns), each of length $\geq t$

$emb_tgt \leftarrow \text{Enc}(X_tgt)$

**foreach** $i \in 1 \ldots |X_sup|$ **do**

  $emb_sup_i \leftarrow \text{Enc}(X_sup_{i,1\ldots t-1})$

**end**

`/* Euclidean distance used */`

$j \leftarrow \arg\min_i \{ \text{dist}(emb_tgt, emb_sup_i),\ i \in 1 \ldots |X_sup| \}$

$y_gen \leftarrow \text{Dec}(emb_tgt)$

$y_ret \leftarrow X_sup_{j,t}$

$cands \leftarrow [\ y_gen,\ y_ret\ ]$

`/* ⊕ denotes concatenation */`

$k \leftarrow \arg\max_i \{ \text{NSP}(\text{Enc}(X_tgt \oplus cands_i)),\ i \in 1 \ldots |cands| \}$

**return** $cands_k$

In our experiments, we found that retrieval baselines are quite effective in the automatic metrics considered. Therefore, we combined retrieval techniques with our response generation model in a hybrid approach – see Algorithm 1.

The retrieval component is set up as follows: when predicting the $t$-th turn of the test dialogue, the model embeds its context of length $t-1$ as well as all the support dialogue contexts of the same length $t-1$ using the fine-tuned dialogue encoder. The encoding for the dialogue context is the hidden state of the last layer of the Transformer model at the position corresponding to the last token in the context. Then, it selects the nearest support context to the target context and picks its $t$-th turn as the retrieved candidate response.

Finally, the model's own generated response and the best retrieved candidate response are ranked using the NSP classification head, i.e. both responses are concatenated with the ground-truth context and the one with the higher $P_{\text{NSP}}$ (Eq. 5.3) is selected. The above steps are visualised in Figure 5.2.

## 5.3 Baselines and Competing Models

We compare our hybrid model to the retrieval baselines provided by the DSTC-8 organisers. The baselines ignore the training data and rely solely on the support sets: they embed each support dialogue's context and find the one nearest to the target context using cosine distance as the metric. They then return the turn following the identified context as the predicted response. There are two retrieval-only baselines, which differ in their encoder: (1) BERT-based (Devlin et al., 2019), taken off-the-shelf, and (2) SentencePiece/FastText-based — representing text as sequences of subword units ('pieces'), with subword tokenisation logic trained in an unsupervised way, in our case on on the Reddit Conversations corpus (the approach is modelled after Gu et al., 2018).

Another baseline provided is a generation-only model, a bidirectional LSTM-based HRED (Serban et al., 2016) trained on MetaLWOz.

All the submissions at the final stage of the challenge are as follows (Li et al., 2020):

— **Team A** trained a BiLSTM on the provided Reddit corpus, then fine-tuned the model at test-time using a mixture of MetaLWOz and MultiWOZ support dialogues, augmented to the context of the target dialogue, and dynamically-sampled Reddit threads,

— **Team B** — the work described in this chapter,

— **Team C** first fine-tuned GPT-2 on the MetaLWOz training corpus, then fine-tuned it further on the support sets of the MetaLWOz and MultiWOZ test sets,

— **Team D** trained a BiLSTM encoder and attentional LSTM decoder on both Reddit and MetaLWOz training corpora, without any fine-tuning to the test sets.

## 5.4 Datasets

We use the main dataset for DSTC-8 Track 2 "Fast Domain Adaptation" MetaLWOz which we described earlier in Section 2.7. Example dialogues from MetaLWOz can be found in Appendix A.

For evaluation purposes, the challenge organisers provide MultiWOZ also described in Section 2.7. MultiWOZ is not present at the base training stage, and a given dialogue model only gets exposed to this data via support dialogues during the adaptation stage, therefore it is used as a means for evaluating the adaptation performance to the data substantially different from the main trainset. Dialogues in MultiWOZ contain NLU annotations, particularly for intent and

| | |
|---|---|
| **Utterance** | I am looking for a particular restaurant . It is called pizza hut city centre . |
| **Markup** | `{'Restaurant-Inform': [['Name', 'pizza hut city centre']]}` |
| **Utterance** | I am looking for a place to to stay that has cheap price range it should be in a type of hotel |
| **Markup** | `{'Hotel-Inform': [['Type', 'hotel'], ['Price', 'cheap']]}` |
| **Utterance** | I would like a taxi from Saint John 's college to Pizza Hut Fen Ditton . |
| **Markup** | `{'Taxi-Inform': [['Dest', 'pizza hut fen ditton'], ['Depart', "saint john 's college"]]}` |
| **Utterance** | Okay that will work . Can you please tell me their phone number , postcode and the entrance fee ? |
| **Markup** | `{'Attraction-Request': [['Fee', '?'], ['Post', '?'], ['Phone', '?']]}` |
| **Utterance** | Just any time after 10:00 , can I get the train ID of one of them please ? |
| **Markup** | `{'Train-Inform': [['Leave', '10:00']], 'Train-Request': [['Id', '?']]}` |

TABLE 5.1: Example annotated utterances from MULTIWOZ

slots, which we use in order to to evaluate the systems' goal-oriented performance — see some example utterances in Table 5.1.

## 5.5 Experimental Setup and Evaluation

We perform training in two stages: training of the base model and fine-tuning it to the target dialogue's support set. At the first stage, we train the model for the maximum of 5 epochs with early stopping. The fine-tuning stage goes on for 1 epoch. GPT-2 models use the context of 3 exchanges, or 5 turns: bot-user-bot-user-bot, predicting the next user's utterance. We mainly used the 'small' GPT-2 checkpoint by HuggingFace — we also tried the 'medium' one, but found no improvement with it in our task.

### 5.5.1 Human Evaluation

| Rank | Submission | Win rate (%) |
|---|---|---|
| 1 | Gold response | 62.32 |
| **2** | **Team B (ours)** | **56.85** |
| 3 | Team C | 52.07 |
| 4 | Team A | 47.35 |
| 5 | Baseline 1 | 44.18 |
| 6 | Team D | 37.34 |

TABLE 5.2: Ranking from judges' pairwise comparisons

The main systems' goal is to generate appropriate responses towards maintaining a natural cooperative dialogue on the user's side, so the main evaluation is performed involving human judges. Specifically, Amazon Mechanical Turk workers were tasked to compare the candidate responses given the dialogue context. Each comparison was pairwise between the results of two systems presented in random order. Judges ranked the responses against the following criteria (Li et al., 2020):

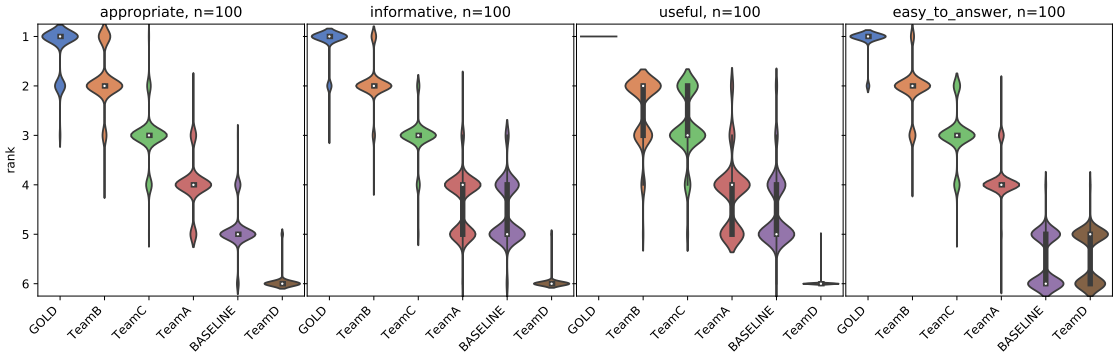



FIGURE 5.3: DSTC-8 Fast Domain Adaptation — human evaluation

— *Usefulness* — whether the response is useful given the dialogue context and the user's overall final goal,

— *Informativeness* — whether the response specifically contains information relevant to the conversation,

— *Appropriateness* — whether the response is appropriate (on-topic, of a reasonable length, not repetitive) to the conversation,

— *Easiness to answer* — given a hypothetical conversational bot on the system side, whether the response will be a valid input for it and presumably straightforward to process.

For each pairing, 3 independent comparisons were performed against each metric. The number of comparisons required was reduced by letting the MULTISORT algorithm (Maystre and Grossglauser, 2017) determine which responses to compare, causing more similar systems with similar performance to be compared more often with each other. Bootstrapping over the 100 randomly chosen dialogue contexts was used to determine average ranks and assess the ranking robustness (Hall et al., 2009).

### 5.5.2 Automatic Evaluation

In addition to human evaluation, we also assess model performance using automatic metrics. The models were evaluated on METALWOZ against word-overlap metrics such as BLEU-1–3, CIDEr, METEOR, ROUGE-L using the NLGEval package (Sharma et al., 2017). Although not ideal for the specifics of dialogue and spoken language in general (Lowe et al., 2017; Dziri et al., 2019), such metrics approximate the overall quality of a response generation model and are especially useful for intermediate evaluation. We evaluate models in two modes on METALWOZ: in *pure task*, support dialogues are drawn from the same domain and task as target dialogue; in *cross-task*, support and target dialogues are from the same domain, but different tasks.

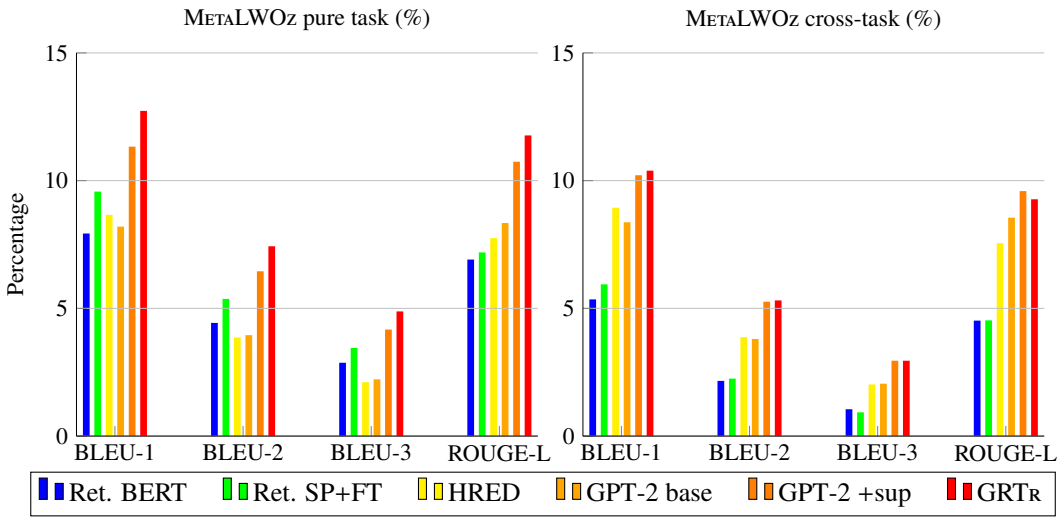



Figure 5.4: Automatic evaluation on MetaLWOz

We also perform additional evaluation of Entity/Intent F1 of the MultiWOZ dataset in pure task mode with pre-trained NLU taggers from the ConvLab package (Lee et al., 2019b). There is no MultiWOZ data available at the first stage (base model training), so all the exposure our model has to this dataset is via support dialogues. Complementary to MetaLWOz evaluation, this stage is designed for assessing the models' goal-oriented performance.

## 5.6 Results and Discussion

### 5.6.1 Human Evaluation

Results of pairwise comparisons are shown in Table 5.2. Our GRTr system's responses (Team B) were preferred by the judges in **56%** of direct comparisons. This surpasses the next best system (Team C) performance by more than **4%**, with only the gold human responses being chosen more frequently.

Furthermore, from the bootstrap ranking distribution (Figure 5.3, lower rank numbers are better), we see that, apart from the gold human responses (blue graphs), our model's outputs (orange graphs) are consistently preferred over other submissions by the judges. Of all metrics used, the most notable are 'appropriateness' and 'usefulness'. On the former, GRTr responses have the second visible peak at rank 1 competing with gold responses. On usefulness however, rank 1 is held by the gold responses with no variation, and our model has the second visible peak at rank 3, thus almost tying with Team C (green graphs).

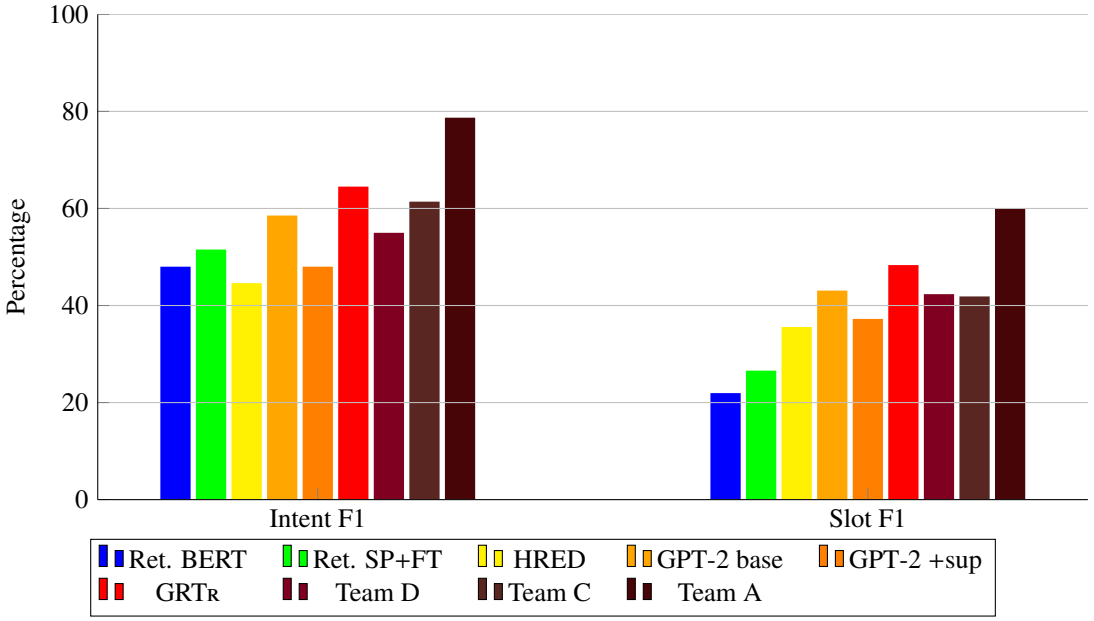



FIGURE 5.5: Automatic evaluation on MULTIWOZ

### 5.6.2 Automatic Evaluation

Results on METALWOZ and MULTIWOZ against automatic evaluation metrics are shown in Figures 5.4 and 5.5, respectively (more detailed METALWOZ evaluation is presented in Tables B.1 and B.2 of Appendix B). We observe that retrieval baselines attain very competitive performance on both datasets, with FastText embeddings from Reddit leading to overall better results than off-the-shelf BERT, especially in the *pure task* setting.

With GRTR, we performed an ablation study to have a closer look into its performance. We evaluated three versions:

— GPT-2 base, a generation-only model trained on METALWOZ and not making use of the support data,

— GPT-2 +sup, the base model fine-tuned to support data, also not using the retrieval logic,

— GRTR, our full hybrid model.

As seen in Figure 5.4, there is strong dependence on support dialogues ('base' vs. '+sup') as the base model mostly struggles to compete with the baselines. Adding retrieval logic ('GRTr' vs. '+sup') results in further performance gains. HRED and GPT-2 base, the two models that did not use support dialogues, had comparable performance on METALWOZ.

In goal-oriented metrics on MULTIWOZ (see Figure 5.5), the same performance pattern is observed with retrieval models, but GPT-2 in the generation-only version performs surprisingly

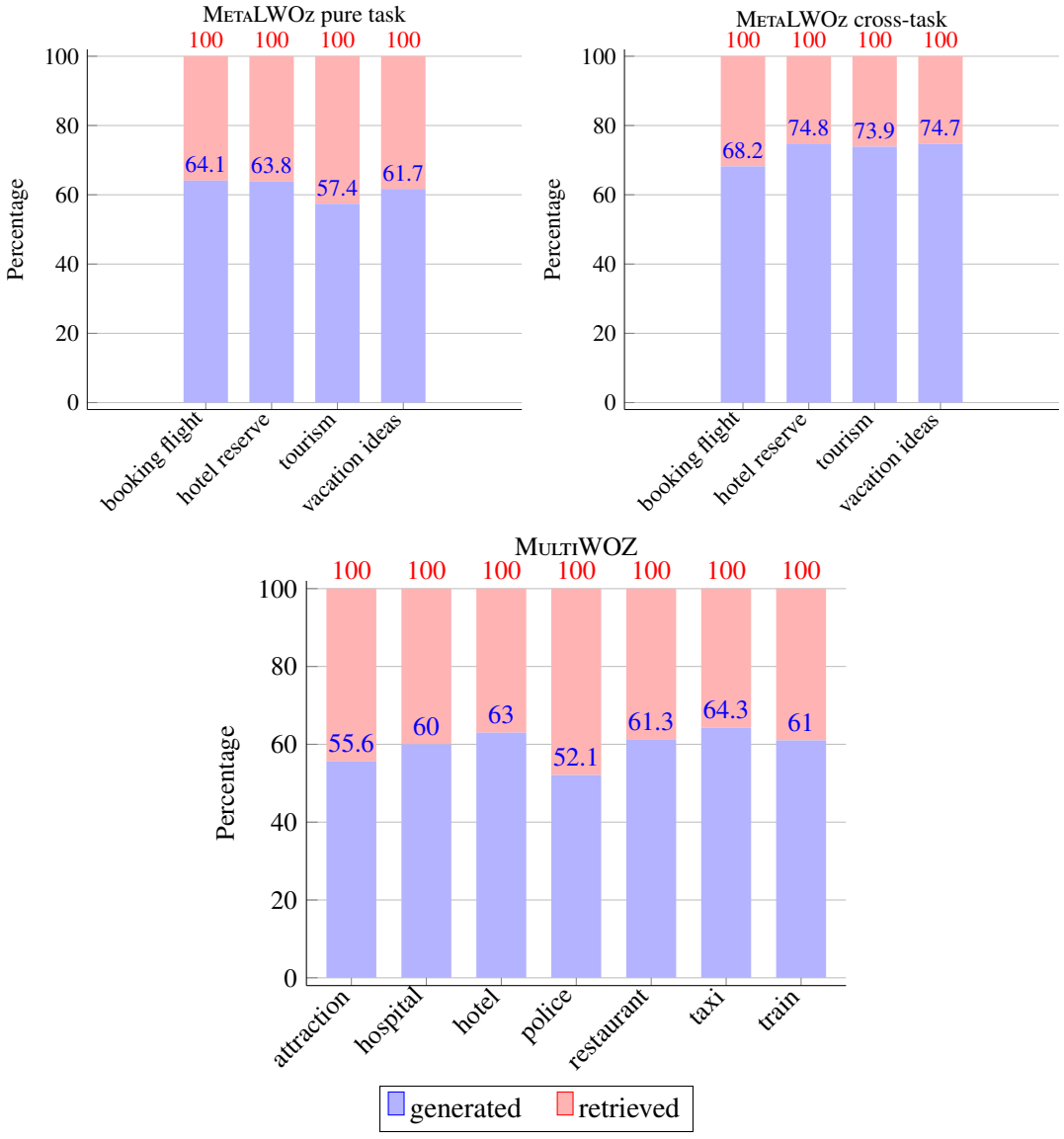



Figure 5.6: GRTr generated/retrieved response ratios

better when not fine-tuned to support set ('base'). On the other hand, the hybrid model experiences even more performance gain than on MetaLWOz. Presumably, generating responses for this dataset is harder due to the fact that it is not represented at the main training stage, and there is not much utterance overlap with MetaLWOz, so little knowledge transfer takes place in this experiment. Compared to other submissions, we observe that GRTr still outperforms most of the competitors and only gives way to Team A's system. We hypothesise here the best MultiWOZ model (Team A) was fitted to the automatic evaluation metrics too tightly, with the negative side effect observable in human evaluation results of Table 5.2 and Figure 5.3, where this system was prevalently ranked $4^{th}$ and $5^{th}$.

### 5.6.3 Analysis of The Generated/Retrieved Responses

| | | |
|---|---|---|
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I need some advice about Rome. |
| | Wiz | What do you want to know? |
| **Gold response** | Usr | I like history, where should I go? |
| **Predicted response** | Usr | Who was the founder of Rome? |
| **Context** | Wiz | Hello, how may I help you? |
| | Usr | I'm looking for a train to broxbourne. |
| | Wiz | Where would you like to leave from, and what day and time would you like to travel? |
| | Usr | I'm going to be leaving from Cambridge. I need to leave on Tuesday by 13:00. |
| | Wiz | Is there a time you would like to get there by? |
| **Gold response** | Usr | I don't have an arrival preference. |
| **Predicted response** | Usr | No I don't have a leave time preference. Just not too early if I don't have too. |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I'd like to book a flight. |
| | Wiz | Okay, any particular place in mind? |
| | Usr | How am I supposed to book a flight to Greece? Me and my kawaii girlfriend were wondering. She says "Ooooo, Greece. :3" |
| | Wiz | I can book a flight for two if you want, There will be an evening flight |
| | Usr | Hmm, wait, I don't really wanna book a flight. We were just curious! She says "Hey! No bookies! :(" |
| | Wiz | Oh, I was confused by that. Can I have your email address so I could send several flight options for you |
| **Gold response** | Usr | Sure thing! My email address is "weeabooking@otaku.corn". She says "I wanna watch my anime now! ;_;" |
| **Predicted response** | Usr | Well, I guess I'll just get back to you. Thanks! |

TABLE 5.3: GRTR example responses

In Figure 5.6, we show per-domain ratios of retrieved/generated responses from the hybrid model. We find that the majority of the responses are generated, and the retrieval logic works as the fallback option. On METALWOZ, which the model had more exposure to during the training, generated responses ratio is generally slightly higher than that on MULTIWOZ which was only seen by the model via support dialogues. Consequently, the model's overall confidence on this dataset is lower, which results in more frequent fallbacks.

Generated candidates rarely duplicate the retrieved ones: we found that the percentage of predictions with identical generated/retrieved candidates is 0.7% (16 in total) for METALWOZ pure, 0.6% (15 in total) for METALWOZ cross, and 0.3% (10 in total) for MULTIWOZ. Also, more detailed information on the distribution of pairwise distances between GRTR response candidates can be found in Figures B.1 — B.3 of Appendix B. Moreover, in Tables B.3 — B.8 of Appendix B, we show GRTR example predictions with the closest generated and retrieved candidates, as well as the most distant ones for METALWOZ pure task, METALWOZ cross-task, and MULTIWOZ datasets. We observe that the generated/retrieved candidates which were scored close to each other are either paraphrases of some generic phrase (e.g. "OK I'll go with that", "I like the idea") or both work well in the dialogue context (see examples for more detail). On the other hand, in

cases with a higher difference between the generated and the retrieved candidate's scores, most of the time we observe that the retrieved one didn't match the context very well.

Overall, we observe in Table 5.3 that there are many cases in the data where the gold response cannot possibly be inferred from the dialogue context. Specifically, the task was posed in the way that no extra data, such as a knowledge base or task description, was provided to the system — therefore, the main goal intended for the hypothetical ideal system is to naturally model human responses in a co-operative goal-oriented dialogue, and to do that in a data-efficient way. This is reflected in the way human judges are asked about response quality.

## 5.7 Conclusion

We presented a hybrid generative-retrieval approach to goal-oriented dialogue with fast domain adaptation via transfer learning. It attains robust and diverse language generation performance across domains, and uses retrieval logic as a fallback mechanism in cases of low confidence. Our method is ranked 1st by human judges of DSTC-8 Fast Domain Adaptation task, and it attains performance superior to a series of baselines in automated metrics on MetaLWOz and MultiWOZ datasets. The future directions of this research mainly include incorporating a more principled fine-tuning technique (i.e. 'learning to learn') and will be discussed in detail in Chapter 9.

# Chapter 6

# Spoken Disfluency Detection

Starting with this chapter, we will focus on practical aspects of data-efficient dialogue modelling. The problem we are going to address here refers back to our study in Chapter 3. As we saw, neural dialogue models lack robustness to certain aspects of spoken language such as high surface variability and the presence of disfluencies (e.g. self-corrections, hesitations, restarts). And although robustness can be improved using more representative training data, it will require datasets of impractical sizes to account for all the linguistic phenomena of interest while keeping the performance on the target downstream task the main priority. From a developer's point of view though, it is highly desirable to be able to develop systems which can be trained from 'clean' examples while also able to generalise to the very diverse disfluent variations on the same data — thereby enhancing both data-efficiency and robustness. Therefore, it can be beneficial for the dialogue systems research to develop a dedicated model that detects such variations in the user's input and passes this information into the downstream dialogue pipeline — either modular or end-to-end.

In this chapter, we present a multitask LSTM-based model[1] for the incremental detection of disfluency structures, which can be hooked up to any component for incremental interpretation (e.g. an incremental semantic parser), or else simply used to 'clean up' the current utterance as it is being produced. We train the system on the Switchboard Dialog Acts (SWDA) corpus and present its accuracy on this dataset. Our model outperforms prior neural network-based incremental approaches by about 10 percentage points on SWDA while employing a simpler architecture. To test the model's generalisation potential to goal-oriented utterances with their characteristic types of disfluency patters, we evaluate the same model on the bAbI+ dataset (presented in Section 3.3), without any additional training. This shows that our approach has good generalisation potential, and sheds more light on which types of disfluency might be amenable to domain-general processing.

[1] Tensorflow (Abadi et al., 2015) and PyTorch (Paszke et al., 2017) implementations together with the trained models are available at http://bit.ly/multitask_disfluency

## 6.1 Motivation

As discussed in Section 2.6.1, spontaneous spoken dialogue is often disfluent, containing pauses, hesitations, self-corrections, and false starts. Processing such phenomena is essential in understanding a speaker's intended meaning and controlling the flow of the conversation. Furthermore, this processing needs to be *word-by-word incremental* to allow further downstream processing to begin as early as possible in order to handle real spontaneous human conversational behaviour.

In this chapter, we build upon the previous best approaches to incremental disfluency detection, the neural models of Hough and Schlangen (2015) and Schlangen and Hough (2017). Our contributions are that: (1) we produce a new multitask LSTM-based model with a simpler architecture for incremental disfluency detection, with significantly improved performance on the SWDA, a disfluency-tagged corpus of open-domain conversations; and (2) we perform a generalisation experiment measuring how well the models perform on unseen data using the controlled environment using bAbI+ (Shalyminov et al., 2017), a dataset containing goal-oriented dialogue utterances with the specific vocabulary and syntactic structures which make it fundamentally different from SWDA.

## 6.2 A Multitask LSTM-based Model for Spoken Disfluency Detection

Our approach to disfluency detection is a sequence tagging model which makes single-word predictions given context words $w_{t-n+1}, ..., w_t$ of a maximum length $n$. We train it to perform two tasks jointly (c.f. Hough and Schlangen, 2015):

1. predicting the disfluency tag of the current word, $P(y_t|w_{t-n+1}, ..., w_t)$, and
2. predicting the next word in the sequence in a language model way, $P(w_{t+1}|w_{t-n+1}, ..., w_t)$.

At training time, we optimise the two tasks jointly, but at test time we only look at the resulting tags and ignore the LM predictions.

Our model uses a shared LSTM encoder Hochreiter and Schmidhuber (1997) with combined 'Word/Part-of-Speech tag' tokens which provides context embedding for two independent multilayer perceptrons (MLPs) making the predictions for the two tasks — see Figure 6.1. The combined token vocabulary (word+POS) size for the SWDA dataset is approximately 30% larger than the original word-only version — given this, concatenation is the simplest and most efficient way to pass POS information into the model.

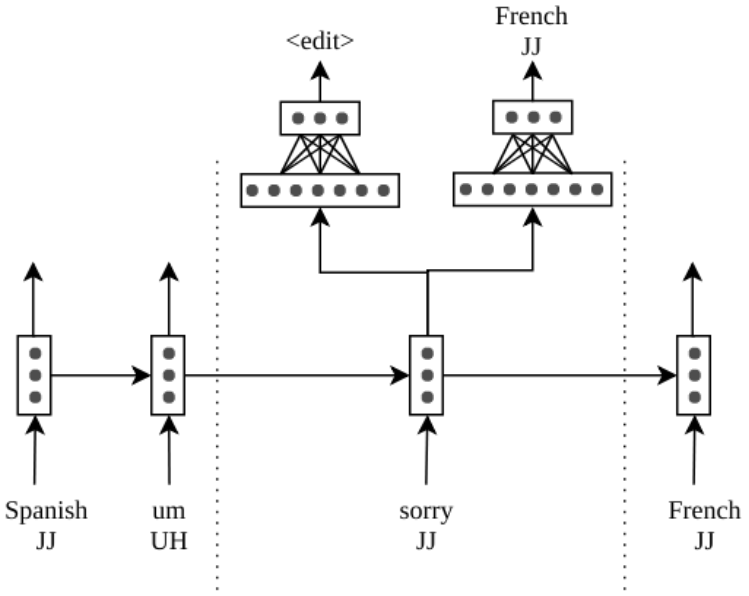



FIGURE 6.1: Multitask LSTM disfluency detector architecture

The intuition behind adding an additional task to optimise for is that it *serves as a natural regulariser*: given an imbalanced distribution of a very few labels (see Section 6.3 for the dataset description), only learning disfluency labels may lead to a higher degree of overfitting, and introducing an additional task with a significantly wider output space can help the model generalise better.

Other potential benefits of having the model work as an LM is the possibility of unsupervised model improvements, e.g. pre-training of the model's LM part from larger text corpora or 1-shot fine-tuning to new datasets with different word sequence patterns.

In order to address the problem of significantly imbalanced training data (the majority of the words in the corpus are fluent), we use a weighted cross-entropy loss in which the weight of a data point is inversely proportional to its label's frequency in the training set. Our overall loss function is of the form:

$$L = WL_{main} + \alpha L_{lm} + \frac{\lambda}{2}\sum_{i} w_i^2 \tag{6.1}$$

where $WL_{main}$ and $L_{lm}$ are respective losses for the disfluency tagging (class-weighted) and language modelling tasks (LM loss coefficient $\alpha$ is tuned empirically). We use class weights in the main task's loss to deal with the highly imbalanced data (see Section 6.3 for an overview od the data), so that the weight of the $k^{th}$ class is calculated as $W_k = 1/(C_k)^{\gamma}$, where $C_k$ is the number of $k^{th}$ class instances in the training set, and $\gamma$ is a smoothing constant set empirically. The last term in Eq. 6.1 is the L2 regularisation which we apply to the model's weight parameters $w_i$ (those

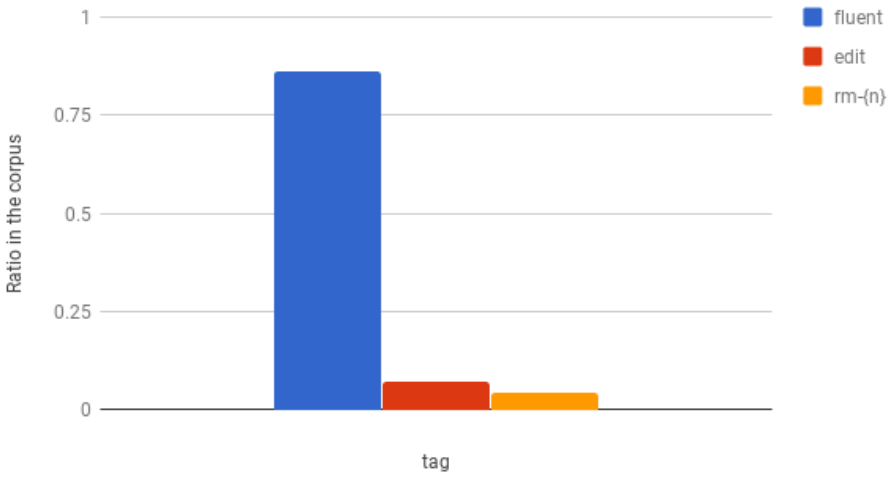



FIGURE 6.2: Statistics of the SWDA corpus

of word embeddings, LSTM gates, and MLPs) leaving all the biases intact. The L2 coefficient $\lambda$ is also tuned empirically.

## 6.3 The Switchboard Dialog Acts Dataset

For training our model, we use the Switchboard Dialog Acts dataset (SWDA) with manually annotated disfluency tags (Meteer et al., 1995). We use a pre-processed version of the dataset by Hough and Schlangen (2015) containing 90,497 utterances with transformed tagging: following their convention, there are 27 tags in total consisting of: <f/> tag for fluent tokens; <e/> for edit tokens; <rm-{n}/> tags for repair tokens that determine the start of the reparandum to be *n* tokens/words back; and <rpSub> & <rpDel> tags which mark the end of the `repair` and classify whether the repair is a *substitution* or *deletion* repair. The latter tokens can be combined with <rm-{n}> tokens, which explains the total of 27 tags — see (6.2) for an example where the `repair` word, 'Spanish', is tagged as <rm-4><rpSub> meaning this is a substitution repair that retraces 4 tokens back from the current token (see Figure 6.2).

$$\underset{\langle f/\rangle}{\text{with}}\ \underbrace{[\text{Italian}\ \underset{\langle f/\rangle}{+}}_{reparandum}\ \underbrace{\{\text{uh no uh}\}}_{\substack{\langle e/\rangle\langle e/\rangle\langle e/\rangle \\ interregnum}}\ \underbrace{\text{Spanish}]}_{\substack{\langle rm{-}4\rangle \\ \langle rpSub\rangle \\ repair}}\ \underset{\langle f/\rangle}{\text{cuisine}} \tag{6.2}$$

The distribution of different types of tokens is highly imbalanced: only about 4% of all tokens are involved in disfluency structures (the detailed statistics are shown in the tables in the end of this chapter). See above, Section 6.2 for how our model deals with this.

## 6.4 Disfluency Detection Generalisation to bAbI+

To evaluate the out-of-dataset generalisation properties of our model and that of Hough and Schlangen (2015), we employ additional data which we generate using bAbI+ tools introduced in Chapter 3. bAbI+ augmentations can be mixed in with complete control over the syntactic and semantic contexts in which the phenomena appear, and therefore the bAbI+ environment allows controlled, focused experimentation of the effect of different phenomena and their distributions on the performance of different models. Here, we use bAbI+ tools to generate new data for the controlled generalisation experiment[2] of what kinds of disfluency phenomena are captured better by each model.

We focus here on the following disfluency patterns:

— *Hesitations*, e.g. as in "we will be *uhm* eight" (mixed in are single edit tokens);

— *Prepositional Phrase restarts (PP-restart)*, e.g. "in a *in a um in a* moderate price range" (repair of a PP at its beginning with or without an interregnum);

— *Clausal restarts (CL-restart)*, e.g. "can you make a restaurant *uhm yeah can you make a restaurant* reservation for four people with french cuisine in a moderate price range" (repair of the utterance from the beginning starting at arbitrary positions);

— *Corrections (NP and PP)*, e.g. "with Italian *sorry Spanish* cuisine", as was initially discussed in Section 6.1.

We generated independent bAbI+ datasets with each disfluency type. The disfluency phenomena above were chosen to resemble disfluency patterns in the original SWDA corpus (see Tables 6.3, 6.5 for examples), as well as intuitive considerations for the phenomena relevant for goal-oriented dialogue (namely, corrections).

The intuition for a generalisation experiment with data like this is as follows: while having similar disfluency patterns, our bAbI+ utterances differ from SWDA in terms of the vocabulary and the word sequences themselves as they are in the domain of goal-oriented human-computer dialogue — this property makes it possible to evaluate the generalisation capabilities of a model outside its training domain.

## 6.5 Evaluation and Experimental Setup

We employ exactly the same evaluation criteria as Hough and Schlangen (2015): micro-averaged F1-scores for edit ($F_e$) and <rm-{n}/> tokens ($F_{rm}$) as well as for whole repair structures ($F_{rps}$).

[2]Data is available at http://bit.ly/babi_plus_disfluencies_study

| **Model** | $\mathbf{F_e}$ | $\mathbf{F_{rm}}$ | $\mathbf{F_{rps}}$ |
|---|---|---|---|
| Hough and Schlangen (2015) | 0.902 | 0.711 | 0.689 |
| Schlangen and Hough (2017) | 0.918 | — | 0.719 |
| LSTM | 0.915 | 0.693 | 0.775 |
| ULMFiT MT-LSTM | 0.889 | 0.735 | 0.795 |
| MT-LSTM | **0.919** | **0.753** | **0.816** |

TABLE 6.1: Evaluation of the disfluency tagging models on SWDA

| **Model** | **hesitations ($\mathbf{F_e}$)** | **PP restarts** | | | **CL-restarts** | | |
|---|---|---|---|---|---|---|---|
| | | $\mathbf{F_e}$ | $\mathbf{F_{rm}}$ | $\mathbf{F_{rps}}$ | $\mathbf{F_e}$ | $\mathbf{F_{rm}}$ | $\mathbf{F_{rps}}$ |
| Hough and Schlangen (2015) | 0.917 | 0.774 | 0.875 | 0.877 | 0.938 | 0.471 | 0.630 |
| LSTM | **0.956** | **1.000** | 0.982 | 0.993 | 0.948 | 0.360 | 0.495 |
| ULMFiT MT-LSTM | 0.902 | 0.917 | 0.937 | 0.963 | 0.958 | 0.456 | 0.637 |
| MT-LSTM | 0.910 | **1.000** | **0.993** | **0.997** | **0.991** | **0.484** | **0.659** |

TABLE 6.2: Controlled generalisation evaluation on the 3 bAbI+ datasets (one per each disfluency phenomenon)

We compare our Multitask LSTM (or MT-LSTM) model to its single-task version (disfluency tag predictions only) as well as to the system of Hough and Schlangen (2015) and the joint disfluency tagging/utterance segmentation model of Schlangen and Hough (2017) on all of the applicable word-level metrics on dialogue transcripts. These use a hand-crafted Markov Model for post-processing, whereas our model works in a streamlined single-stage way.

Apart from that, we evaluate a version of our model that makes use of pre-trained contextual embeddings — namely, AWD-LSTM (Merity et al., 2018) trained on WikiText-103 dataset (Merity et al., 2017) following the ULMFiT technique (Howard and Ruder, 2018)[3]. Among the rest of widely-used pre-trained contextual embeddings (e.g. ELMo, BERT, GPT-2), AWD-LSTM is a strictly left-to-right unidirectional model with the possibility of incremental word-by-word state updates which is an essential feature of all the models in this experiment. In order to make use of the pre-trained embeddings, we don't use combined word-POS tokens with this model and use separate LSTM encoders instead — the pre-trained one for words and a from-scratch one for POS-tags, respectively.

We train our model using the SGD optimiser and monitor the $F_{rm}$ on the dev set as a stopping criterion. The model's hyperparameters are tuned heuristically, the final values are listed in the tables in the end of this chapter.

[3]We used a pre-trained model from `fast.ai` — code available at http://tiny.cc/ulmfit_disfluency

| Repair | Freq. |
|---|---|
| i i *i* | 139 |
| the the *the* | 33 |
| and and *and* | 31 |
| it it *it* | 29 |
| its its *its* | 26 |
| it was *it was* | 67 |
| i dont *i dont* | 57 |
| i think *i think* | 44 |
| in the *in the* | 39 |
| do you *do you* | 23 |
| a lot of *a lot of* | 7 |
| that was *uh that was* | 5 |
| it was *uh it was* | 5 |
| what do you *what do you* | 4 |
| i i dont *i dont* | 4 |

TABLE 6.3: Most common repairs in SWDA of length 1—3

| POS pattern | Examples | Freq., % |
|---|---|---|
| DT NN DT NN | this woman this socialite<br>a can a garage<br>the school that school | 0.10 |
| JJ NN JJ NN | high school high school<br>good comedy good humor<br>israeli situation palestinian situation | 0.03 |
| DT UH DT NN | that uh that punishment<br>the uh the cauliflower<br>that uh that adjustment | 0.02 |
| DT NN UH DD NN | a friend uh a friend<br>a lot uh a lot<br>a lot um a lot | 0.01 |
| NN PRP VBP NN NN | ribbon you know hair ribbon<br>thing you know motion detector | 0.01 |

TABLE 6.4: SWDA repairs by POS-tag pattern

## 6.6 Results

The results are shown in Table 6.1. Both single- and multitask LSTM are able to outperform the Hough and Schlangen (2015) model on edit tokens and repair structures, but the multitask one performs significantly better on <rm-{n}/> tags and surpasses both previous models. The reason $F_{rps}$ is higher than $F_{rm}$ in general is that due to the tag conversion, fluent tokens inside reparandums and repairs are treated as part of repair, and they contribute to the global positive and negative counters used in the micro-averaged F1. Large-scale pre-training, while potentially useful for better coverage of the target dataset, did not affect the final disfluency detection accuracy as ULMFiT MT-LSTM's results show. The reason for that could be the initially large size of the underlying pre-trained AWD-LSTM: it contains 3 layers with 1024, 1024, and 400 neurons, respectively, which is significantly more than what our best performing model uses (we also tried only using the first layer of the AWD-LSTM, which didn't result in any significant accuracy improvement).

Controlled generalisation experiment results are shown in Table 6.2 — note that we could only run the model of Hough and Schlangen (2015) on bAbI+ data because that of Schlangen and Hough (2017) works in a setup different from ours. It can be seen that the LSTM tagger is somewhat overfitted to edit tokens on SWDA. This is the reason it outperforms the Multitask LSTM on the hesitations dataset and has a tied 1.0 on edit tokens on PP restarts dataset. In all other cases, Multitask LSTM demonstrates superior generalisation.

As for NP/PP self-corrections which are not present in Table 6.2: none of the systems tested were able to handle these. Evaluation on this dataset revealed 0.0 accuracy with all systems. We discuss these results below.

| **Keyword pattern** | **Examples** | **Freq., %** |
|---|---|---|
| sorry<e/> * | or *im sorry* no<br>*um im sorry* what<br>thank you *im sorry* i just got home from work | 0.02 |
| sorry<e/> *<rm-*/> | and he told us theres two sixteen bit slots and two eight bit<br>*sorry two four sixteen bit slots and two eight bit* slots available for the user | 0.009 |
| i<e/> mean<e/> * | i mean<br>i mean yeah<br>i mean uh<br>i mean i | 4 |
| i<e/> mean<e/> *<rm-*/> | i mean i i<br>but i mean whats whats happened here is is is<br>i mean you youve | 0.5 |

TABLE 6.5: SWDA repairs by interregnum

## 6.7 Conclusion

We have presented a multitask LSTM-based disfluency detection model which outperforms previous neural network-based incremental models while being significantly simpler than them.

We have also demonstrated the generalisation potential of a disfluency detection model by cross-dataset evaluation. As the results show, all models achieve reasonably high generalisation level on the very local disfluency patterns such as hesitations and PP restarts. However, the accuracy drops significantly on less restricted restarts spanning arbitrary regions of utterances from the beginning. On the majority of the disfluency patterns, our model achieves a superior generalisation level.

Interestingly, none of the models were able to detect NP or PP corrections such as those often glossed in disfluency papers (e.g. "A flight to Boston uh I mean to Denver"). The most likely explanation for this could be the extreme sparsity of such disfluencies in the SWDA dataset.

We performed analysis of SWDA disfluencies in order to explore this hypothesis and examined their distribution based on length in tokens and POS-tag sequence patterns of interest. As shown in Tables 6.3 and 6.4, the vast majority of disfluencies found are just repetitions without speakers actually correcting themselves. This observation is in line with prior studies, showing that the distribution of repair types varies significantly across domains (Colman and Healey, 2011), modalities (Oviatt, 1995), and gender & age groups (Bortfeld et al., 2001) — see Purver et al. (2018) for a nice discussion. While this is very likely the correct explanation, we cannot rule out the possibility that such self-corrections are inherently more difficult to process for particular models —- that needs a separate training dataset that holds frequency of particular repair structures constant.

Incremental phenomena of spontaneous spoken language are one aspect of natural input which dialogue systems should process robustly. Another natural phenomenon that a dialogue system can be exposed to is the presence of out-of-domain (OOD) utterances at the input. With most of the neural dialogue models' logic learned form data and not directly observable, it is often the case that the system's behaviour on unseen input is unpredictable which significantly limits such system's practical usability. In the next chapter, we will continue our work on dialogue systems' robustness by addressing the problem of OOD user's input detection and handling in a predictable way, while keeping the overall training setup data-efficient.

# Chapter 7

# Improving Out-of-Domain Robustness of Dialogue Systems

In this chapter, we are going to continue with the problem of the dialogue systems' robustness to the natural and diverse user's input. In the previous chapter, we worked on improving robustness to surface variations of the user's input represented as the disfluencies in the spoken language. Here, we are going to cover another kind of input that can be considered anomalous, especially for the systems trained in a data-efficient way from a few example dialogues — out-of-domain (OOD) input, i.e. user queries that the system is unable to interpret and process correctly. We explore the problem of OOD robustness of dialogue systems and the associated to it trade-off in accuracies on seen and unseen data.

We present a new experimental testbed for studying the robustness of dialogue systems to OOD input, which is bAbI Dialog Task 6 (Bordes et al., 2017) augmented with OOD content in a controlled way. We then present turn dropout, a simple yet efficient technique for training OOD-robust dialogue systems based on automatic data augmentation and thus alleviating the problem of the dependence on real OOD data. With also propose a simple unified way to train the target systems for OOD input detection as well as for their main task at the same time.

Moreover, based on prior approaches to efficient OOD input detection using autoencoders (see our discussion in Section 2.6.2), we propose a variant of Hybrid Code Network (HCN) dialogue management model (Williams et al., 2017) augmented with an autoencoder for contextual OOD input detection called AE-HCN(-CNN). We experiment with this model, as well as with a range of HCN-family models: Variational HCN (VHCN), Hierarchical HCN (HHCN), the original HCN — adapted for training with turn dropout, and demonstrate the resulting performance improvement as seen on bAbI Dialog Task 6 and Google multi-domain dialogue datasets (Shah et al., 2018).

| | | |
|---|---|---|
| 1 | *sys:* | Hello, welcome to the Cambridge restaurant system. You can ask for restaurants by area, price range or food type. How may I help you? |
| 2 | *usr:* | **I'd like to get away from Monday, August 15, 2016 to Wednesday, August 31, 2016. My budget is $ 3200 and I would leave from Detroit.** |
| 3 | *sys:* | Sorry I didn't catch that. Could you please repeat? |
| 4 | *usr:* | **Will there be frost in Durham next week?** |
| 5 | *sys:* | Sorry I didn't catch that. Could you please repeat? |
| 6 | *usr:* | ***So sorry man*** I want a moderately priced restaurant in the west part of town |
| 7 | *sys:* | <api call> |

TABLE 7.1: Augmented dialogue example (OOD content in bold, segment-level in italics)

## 7.1 The Experimental Environment for OOD Robustness Studies

In order to study the effect of OOD input on end-to-end dialogue system's performance, we created an experimental testbed based on (1) a dataset of real human-computer goal-oriented dialogues and (2) the data augmentation technique — bAbI tools (presented in Chapter 3) for mixing real user utterances from other domains into the main dataset in a controlled way.

As our main dataset, we use bAbI Dialog Task 6 (Bordes et al., 2017) with real human-computer conversations in the restaurant search domain initially collected for the Dialog State Tracking Challenge 2 (Henderson et al., 2014a). Our OOD augmentations are as follows:

— *turn-level* OOD: user requests from a foreign domain — the desired system behaviour for such input is the fallback action,

— *segment-level* OOD: interjections in the user in-domain requests — treated as valid user input and is supposed to be handled by the system in a regular way.

These two augmentation types reflect a specific dialogue pattern of interest (see Table 7.1): first, the user utters a request from another domain at an arbitrary point in the dialogue (each turn is augmented with the probability $p_{ood_start}$), and the system answers accordingly. This may go on for several turns in a row — each subsequent turn is augmented with the probability $p_{ood_cont}$. Eventually, the OOD sequence ends up and the dialogue continues as usual, with a segment-level OOD of the user affirming their mistake. For this study, we set $p_{ood_start}$ to 0.2 and $p_{ood_cont}$ to 0.4[1].

While we introduce the OOD augmentations in a controlled programmatic way, the actual OOD content is natural. The turn-level OOD utterances are taken from conversational datasets in several foreign domains:

— Frames dataset (Asri et al., 2017) — travel booking (1198 utterances),

[1]We experimented with other values of $p_{ood_start}$ and $p_{ood_cont}$ but did not see significant differences in the results. Further experiments for different domains are encouraged using the tools provided

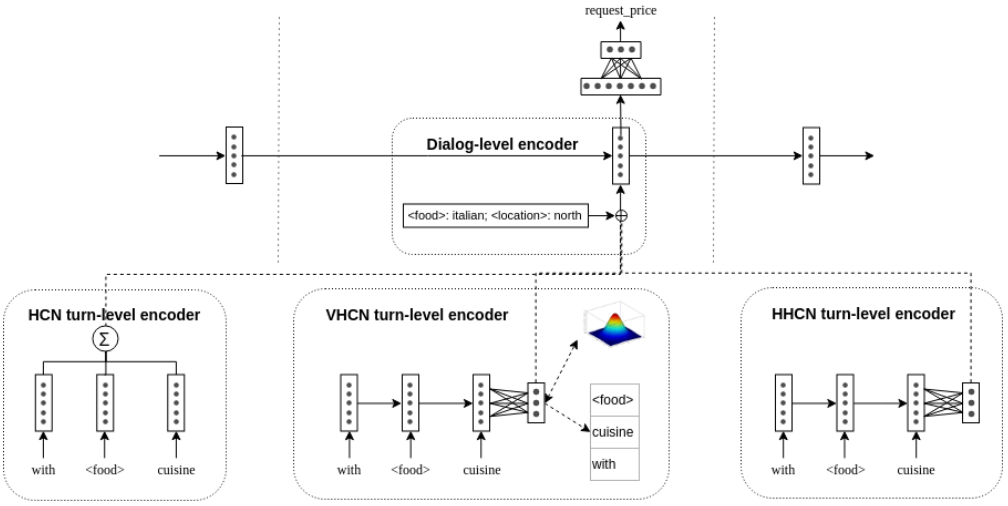



FIGURE 7.1: Hybrid Code Network model family

— Stanford Multi-Domain (SMD) Dialogues Dataset (Eric et al., 2017) — calendar scheduling, weather information retrieval, city navigation (3030 utterances),

— Dialog State Tracking Challenge 1 (Williams et al., 2013) — bus information (968 utterances).

In order to avoid incomplete/elliptical phrases, we only took the first user's utterances from the dialogues.

For segment-level OOD, we mined utterances with the explicit affirmation of a mistake from Twitter and Reddit Conversations datasets (e.g. "my mistake", "I'm so sorry") — 701 and 500 utterances respectively. Our datasets, as well as the tools for the OOD-augmentation of arbitrary datasets of interest are openly available[2].

## 7.2 OOD-Robust Dialogue Models

### 7.2.1 Hybrid Code Network Model Family

In this chapter, we experiment with the Hybrid Code Network family of models (Williams et al., 2017) on bAbI Dialog Task 6 data (Bordes et al., 2017). HCN is reported to be the best performing model to date for the original, IND-only bAbI Dialog Task 6 data — thus, this is our primary experimental setup here.

As shown in Figure 7.1, HCN considers a dialogue as a sequence of turns. At each turn, HCN takes a tuple $(x_t, a_{t-1}, s_t)$ as the input to produce the next system action[3] $a_t$, where $x_t$ is a user

[2]See https://github.com/ishalyminov/ood_robust_hcn
[3]A system action can be either a text output or an api call.

utterance consisting of $N$ tokens, i.e., $x_t = \{x_{t,1}, \ldots, x_{t,N}\}$, $a_{t-1}$ a one-hot vector encoding the previous system action, and $s_t$ a contextual feature vector generated by domain-specific code. The user utterance is encoded as a concatenation of a bag-of-words representation and an average of word embeddings of the user utterance:

$$u_t = \left[ \text{bow}(x_t); \frac{1}{N} \sum_{i=1}^{N} e(x_i) \right] \tag{7.1}$$

where $e(\cdot)$ denotes a word embedding layer initialised with GloVe (Pennington et al., 2014) or Google News-based Word2Vec (Mikolov et al., 2013) — frozen at the training time. HCN then considers the input tuple $(u_t, a_{t-1}, s_t)$ to update the dialogue state through an LSTM (Hochreiter and Schmidhuber, 1997):

$$h_t = \text{LSTM}\left(h_{t-1}, [u_t; a_{t-1}; s_t]\right) \tag{7.2}$$

Finally, a distribution over system actions is calculated by a dense layer with linear projection parameters $W$ and $b$ — the weight matrix and the bias vector, respectively — and with a softmax activation:

$$P(a_t) = \text{softmax}(Wh_t + b) \tag{7.3}$$

Thus, HCN is a hierarchical dialogue management model with a turn-level and a dialogue-level components (we will call them both encoders). The turn-level encoder produces a latent representation of a single dialogue turn, and the dialogue-level encoder augments it with additional dialogue-level features.

We will first explore a series of HCN variations differing in their turn-level encoder (and the corresponding optimisation objective) — see the illustration in Figure 7.1.

The original HCN as described above. Its optimisation objective for a single prediction is the categorical cross-entropy with respect to the log-likelihood of the output dialogue action (here and in Eq. 7.7 we show maximisation objectives for simplicity. In the actual implementation, they are minimised with their sign reversed):

$$\mathcal{L}_{\text{HCN}} = \log p\left(a_t \mid x_1, \ldots, x_t, a_{t-1}, s_t\right) \tag{7.4}$$

where $a_t$ is the dialogue action, $x_1, \ldots, x_t$ is the dialogue context ending with the current user's turn, $a_{t-1}$ is the last system's action, and $s_t$ is the domain-specific dialogue feature vector. Please note, losses of all our HCN models are defined with respect to the model parameters $\theta_{\text{HCN}}$, i.e.

the parameters of the underlying LSTMs, embeddings, and projection layers, which we omit for visual simplicity.

Hierarchical HCN (HHCN) uses an RNN (in our case an LSTM cell) in order to encode each utterance:

$$u_t^{\text{HHCN}} = \text{LSTM}\left(e(x_t)\right) \tag{7.5}$$

The optimisation objective is the same as of HCN. Variants of this model were described by Lee (2017) and Liang and Yang (2018).

Variational HCN (VHCN) uses a Variational Autoencoder as the turn-level encoder, so that the resulting turn encoding is VAE's latent variable $z$ (see also our discussion of latent variable models in Section 2.3.3):

$$u_t^{\text{VHCN}} = \mu\left(\text{LSTM}\left(e(x_t)\right)\right) + \sigma\left(\text{LSTM}\left(e(x_t)\right)\right) * N(0,1) \tag{7.6}$$

where $\mu$ and $\sigma$ are MLPs for predicting $z$'s posterior distribution parameters, and $N(0,1)$ is a sample from its prior distribution, a standard Gaussian (Bowman et al., 2016).

This model differs from the previous two in the fact that it learns dialogue management and input autoencoding jointly. In order to keep the user's input reconstruction task less complex than the main one, we represent VAE's reconstruction targets as bags of words (BoW) instead of sequences — in that, we follow (Zhao et al., 2017). Thus, VHCN optimisation objective is as follows:

$$\mathcal{L}_{\text{VHCN}} = \mathbb{E}_{q(z)}\left[log(p(a_t \mid x_1, \ldots, x_t, a_{t-1}, s_t))\right] + \mathbb{E}_{q(z)}\left[p(x_t^{\text{BoW}} \mid z)\right] - KL\left(q(z \mid x_t) \parallel p(z)\right) \tag{7.7}$$

In the above formula, the first term is the main task's log-likelihood of the dialogue action $a_t$, the second one is the VAE's reconstruction term for the last user's input in the bag-of-words form $x_t^{\text{BoW}}$, and the last turn is $KL$-divergence between the prior and posterior distribution of the VAE's latent variable $z$ — following Bowman et al. (2016), we compute it in a closed form.

Another benefit of the BoW loss is, as reported in Zhao et al. (2017), it helps keep the variational properties of the model (i.e. non-zero KL-term) without the necessity of using the KL-term annealing trick (Bowman et al., 2016) which is itself challenging to control in practice. Unlike the authors of the original BoW loss approximating the presence of each word in the reconstructed bag with a feed-forward neural network and then summing up each word's log-probability in the final loss, we use a simpler method and represent the BoW as a single vocabulary-sized vector

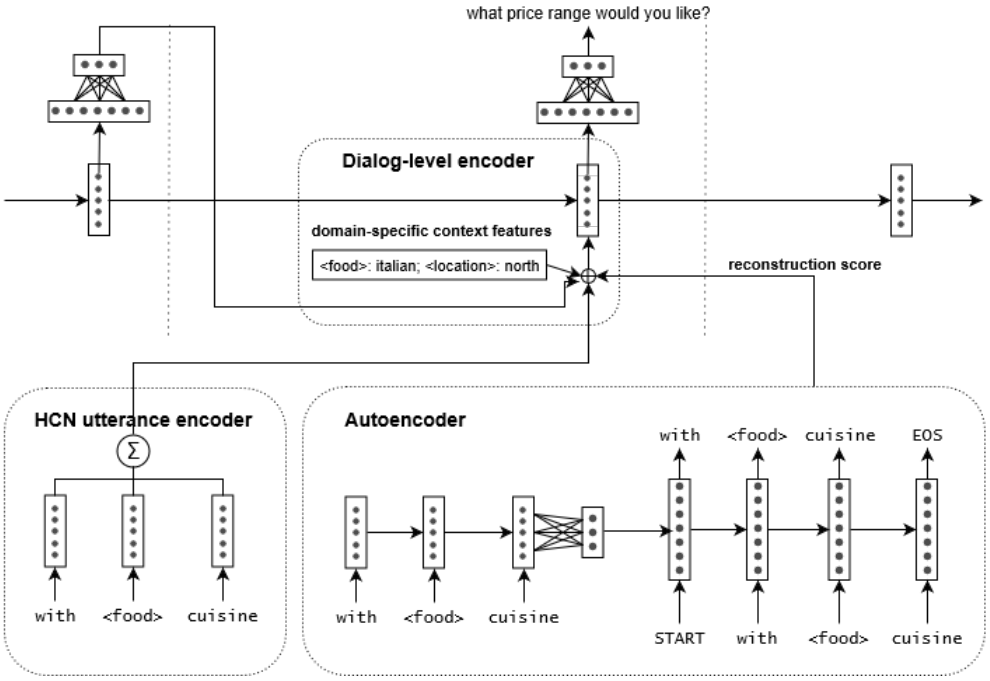



FIGURE 7.2: AE-HCN model architecture

with k-hot values (1 for every word present in the bag, the rest of the elements being 0's), and use a single sigmoid cross-entropy loss for it.

All the models above use the same dialogue-level LSTM encoder with additional features concatenated to the turn representations: BoW turn features, dialogue context features, and the previous system action[4].

### 7.2.2 AE-HCN

Finally, we introduce the two autoencoder-augmented architectures, AE-HCN & AE-HCN-CNN, where the HCN model is aware of the AE's reconstruction score — together with the training method based on automatic input augmentation.

AE-HCN is an HCN whose dialogue-level encoder takes an additional input for dialogue state update — specifically, the autoencoder's reconstruction score $r_t$ for the user's utterance (Figure 7.2):

$$h_t = \text{LSTM}\left(h_{t-1}, [u_t; a_{t-1}; s_t; r_t]\right) \tag{7.8}$$

The autoencoder is a standard SEQ2SEQ model which projects a user utterance into a latent vector and reconstructs the user utterance. Specifically, the encoder reads $x_t$ using a GRU (Cho et al., 2014a) to produce a 512-dimensional hidden vector $h_N^{\text{enc}}$ which in turn gets linearly projected to

[4]Without the loss of the architecture generality, we have action mask vectors as the additional features for the dialogue-level LSTM (Williams et al., 2017), but they do not convey any information and are always set to 1's

a 200-dimensional latent vector $z$ with the corresponding weight and bias parameters $W_z$ and $b_z$, respectively — see the formulas below:

$$h_n^{\text{enc}} = \text{GRU}_{enc}\left(h_{n-1}^{enc}, e(x_n)\right), 1 < n < N \tag{7.9}$$

$$z = W_z h_N^{enc} + b_z \tag{7.10}$$

The output of the decoder $y$ at step $n$ is a distribution over words:

$$P_{\text{dec}}(y_n) = \text{softmax}\left(W_{\text{dec}} h_n^{\text{dec}} + b_{\text{dec}}\right) \tag{7.11}$$

$$h_n^{\text{dec}} = \text{GRU}_{\text{dec}}\left(h_{n-1}^{\text{dec}}, e(y_{n-1})\right) \tag{7.12}$$

$$h_0^{\text{dec}} = W_{\text{dec}} z + b_{\text{dec}} \tag{7.13}$$

where $\text{GRU}_{\text{dec}}$ has 512 hidden units.

The reconstruction score $r_t$ is the normalised generation probability of $x_t$ (which is both the input and the output of the autoencoder):

$$r_t = \frac{\sum_{n=0}^{N} \log P_{dec}(x_n)}{N} \tag{7.14}$$

### 7.2.3 AE-HCN-CNN

AE-HCN-CNN is a variant of AE-HCN where user utterances are encoded using a CNN layer with max-pooling (following Kim, 2014) rather than equation 7.1:

$$x_t = \text{Pooling}_{max}\left(\text{CNN}(e(x_1), ..., e(x_n))\right) \tag{7.15}$$

The CNN layer considers two kernel sizes (2 and 3) and has 100 filters for each kernel size.

## 7.3 Training with Turn Dropout

In order to train a system robust to OOD in the absence of real OOD examples, we employ a negative sampling-based approach and generate them synthetically from the available IND data with a technique we call turn dropout. Namely, we replace random dialogue turns with the synthetic ones, and assign them the fallback action.

| **Hyperparameter** | **HCN** | **HHCN** | **VHCN** |
|---|---|---|---|
| Embedding size | 64 | 128 | 128 |
| Latent variable size | — | — | 8 |
| Turn dropout ratio | 0.4 | 0.6 | 0.3 |
| Learning rate | 0.001 (all models) | | |
| Early stopping threshold | 20 epochs (all models) | | |
| Optimiser | Adam (all models) | | |
| Word dropout ratio | 0.2 (all models) | | |

TABLE 7.2: Model hyperparameters

| | **bAbI Dialog Task 6** | **bAbI Dialog Task 6 + OOD** | | | |
|---|---|---|---|---|---|
| | Overall acc. | Overall acc. | Seg. OOD acc. | OOD acc. | OOD F1 |
| HCN | 0.557 | 0.438 | **0.455** | 0.0 | 0.0 |
| HHCN | 0.531 | 0.418 | 0.424 | 0.0 | 0.0 |
| VHCN | 0.533 | 0.413 | 0.413 | 0.0 | 0.0 |
| TD-HCN | 0.563 | **0.575** | 0.257 | **0.754** | **0.743** |
| TD-HHCN | 0.505 | 0.455 | 0.435 | 0.274 | 0.418 |
| TD-VHCN | **0.565** | 0.545 | 0.407 | 0.530 | 0.667 |

TABLE 7.3: Evaluation results

### 7.3.1 TD-HCN

More formally, our dialogue features are as follows: $\langle f_turn, f_ctx, f_mask, a \rangle$, i.e. turn features (token sequences), dialogue context features, action masks, and target actions respectively.

Under turn dropout, for a randomly selected dialogue $i$ and its turn $j$, we replace $f_turn_{ij}$ with a sequence of random vocabulary words (drawn from a uniform distribution over the vocabulary) and UNK tokens, and corresponding $a_{ij}$ with the fallback action, and leave all other features intact. In this way, we're simulating anomalous turns for the system given usual contexts (as stored in the dialogue RNN's state), and we put minimum assumptions on the synthesised turns' structure (we only limit their lengths to be within the bounds of real utterances).

### 7.3.2 Training the AE-HCN(-CNN)

While essentially similar to the TD-HCN algorithm described above, training these two architectures with data augmentation involves providing reconstruction scores for the 'dropped out' turns. We describe the full procedure below.

To endow an AE-HCN(-CNN) model with a capability of detecting OOD utterances and producing fallback actions without requiring real OOD data, we augment training data with counterfeit

| Dataset | # Dialogues | Avg. #turns per dialogue | # actions |
|---|---|---|---|
| bAbI6 train | 1618 | 20.08 | 58 |
| dev | 500 | 19.30 | 58 |
| test | 1117 | 22.07 | 58 |
| test-OOD | 1117 | 27.27 | 59 |
| GR train | 1116 | 9.07 | 247 |
| dev | 349 | 6.53 | 247 |
| test | 775 | 6.87 | 247 |
| test-OOD | 775 | 9.01 | 248 |
| GM train | 362 | 8.78 | 194 |
| dev | 111 | 9.14 | 194 |
| test | 252 | 8.73 | 194 |
| test-OOD | 252 | 11.25 | 195 |

TABLE 7.4: bAbI6, GR, and GM dataset statistics

turns. We first select arbitrary turns in a dialogue at random according to a counterfeit OOD probability $\rho$, and insert counterfeit turns before the selected turns. A counterfeit turn consists of a tuple $(x_t, a_{t-1}, s_t, r_t)$ as input and a fallback action $a_t$ as output. We copy $a_{t-1}$ and $s_t$ of each selected turn to the corresponding counterfeit turns since OOD utterances do not affect previous system action and feature vectors generated by domain-specific code. Now we generate a counterfeit $x_t$ and $r_t$. Since we do not know OOD utterances a priori, we randomly choose one of the user utterances of the same dialogue to be $x_t$. This helps the model learn to detect OOD utterances because a random user utterance is contextually inappropriate just like OOD utterances are. We generate $r_t$ by drawing a sample from a uniform distribution, $U[\alpha, \beta]$, where $\alpha$ is the maximum reconstruction score of training data and $\beta$ is an arbitrary large number. The rationale is that the reconstruction scores of OOD utterances are likely to be larger than $\alpha$ but we do not know what distribution the reconstruction scores of OOD turns would follow. Thus we choose the most uninformed distribution, i.e. a uniform distribution so that the model may be encouraged to consider not only reconstruction score but also other contextual features such as the appropriateness of the user utterance given the context, changes in the domain-specific feature vector, and what action the system previously took.

## 7.4 Experiment 1: HHCN & VHCN

We train our models only using the original bAbI Dialog Task 6 dataset, and evaluate them on our OOD-augmented versions of it: we use the per-utterance accuracy as our main evaluation metric; the models are trained with the same hyperparameters (where applicable) listed in Table 7.2. The models use the common unified vocabulary including all words from our datasets (including OOD content). The intuition behind this is as follows: production dialogue models

often use word embedding matrices with vocabularies significantly exceeding that of the training data in order to take advantage of additional generalisation power via relations like synonymy, hyponymy, or hypernymy which are efficiently modelled by distributed word representations. Therefore, simply mapping every unseen word to an 'UNK' does not quite reflect that setting.

We tuned our models' hyperparameters using a 2-stage grid search, tracking the development set accuracy. At the first stage, we adjusted the embedding dimensionality of our models (and the latent variable size in case of VHCN). Then, given the values found, at the second stage we adjusted turn dropout ratio at the interval [0.05 – 0.7]. Exact hyperparameter values are detailed in Table 7.2.

The results are shown in Table 7.3 — please note, apart from the accuracies we report OOD F1-measure, a metric showing the model's performance as a conventional OOD detector, with positive class being the fallback action, and the negative being all the IND classes actions.

Finally, given the stochastic nature of VHCN, we reported its mean accuracy scores over 3 runs (we used the same criterion for selecting the best model during the training procedure).

### 7.4.1 Results and Discussion

As our experiment showed, while learning to handle both IND and OOD input with access to IND-only data at the training time, there appears the following trade-off: a model performing better on the 'clean' test turns is prone to lower accuracy on OOD — it can be said that it slightly overfits to its devset. On the other hand, a model regularised with turn dropout during training naturally performs better on unseen OOD turns, but with not as high accuracy on its 'clean', IND test data. Another side of the trade-off is the accuracy of OOD detection vs robust handling of IND input with segment-level noise. As our results showed, the models specifically trained for OOD detection all demonstrate lower accuracy on the noisy IND.

Among the models we evaluated, it is worth noting that the original HCN demonstrated the best performance as an OOD detector (more than **74%** F1-score) and thus overall IND + OOD accuracy on the augmented dataset — more than **57%**. While some parts of its architecture (e.g. mean vector-based turn encoding or bag-of-words feature vector at the utterance level) may not seem to be the most robust solution, the model demonstrate superior overall performance. Averaging at the turn level instead of recurrent encoding (the case of HHCN and VHCN) makes the model less dependent on actual word sequences seen during training but on the keywords themselves.

In turn, VHCN demonstrated superior performance on IND data when trained with turn dropout, more than **56%** — it benefited in terms of both overall accuracy and the absence of false-positive OODs thus improving upon the original HCN's results Williams et al. (2017). An

| | bAbI6 | | | GR | | | GM | | |
|---|---|---|---|---|---|---|---|---|---|
| | Test | Test-OOD | | Test | Test-OOD | | Test | Test-OOD | |
| | P@1 | P@1 | OOD F1 | P@3 | P@3 | OOD F1 | P@3 | P@3 | OOD F1 |
| HCN | 53.41 | 41.95 | 0 | **58.89** | 41.65 | 0 | 41.18 | 27.08 | 0 |
| AE-HCN-Indep | 31.29 | 41.06 | 48.68 | 51.90 | 55.42 | 71.52 | 31.12 | 42.78 | 64.35 |
| AE-HCN | 53.58 | 55.04 | **73.41** | 56.97 | 58.90 | 74.67 | 40.61 | 48.59 | **69.31** |
| AE-HCN-CNN | **55.04** | **55.35** | 70.38 | 58.32 | **64.51** | **81.33** | **45.12** | **52.79** | 68.59 |

TABLE 7.5: AE-HCN(-CNN) Evaluation results

additional challenge was to train it while keeping its variational properties (i.e. reasonably high KL term) — the BoW reconstruction loss which we used in order to simplify the secondary task, helped with this as well (Zhao et al., 2017). On the other hand, while achieving superior performance on clean data, VHCN's properties did not result in OOD handling improvements.

## 7.5 Experiment 2: AE-HCN(-CNN)

To study the effect of OOD input on these two dialogue systems' performance, we use three task-oriented dialogue datasets: bAbI6 as in the previous experiment, as well as GR and GM taken from Google multi-domain dialogue datasets (Shah et al., 2018). Basic statistics of the datasets are shown in Table 7.4. bAbI6 deals with restaurant finding tasks, GM buying a movie ticket, and GR reserving a restaurant table, respectively. We generated distinct action templates by replacing entities with slot types and consolidating based on dialogue act annotations. We augment test datasets (denoted as Test-OOD in Table 7.4) with the procedure described in Section 7.1.

### 7.5.1 Experimental Setup and Evaluation

We comparatively evaluate four different models:

1. an HCN model trained on in-domain training data;
2. an AE-HCN-Indep model which is the same as the HCN model except that it deals with OOD utterances using an independent autoencoder-based rule to mimic Ryu et al. (2017) — when the reconstruction score is greater than a threshold, the fallback action is chosen (we set the threshold to the maximum reconstruction score of training data);
3. an AE-HCN(-CNN) model trained on training data augmented with counterfeit OOD turns — the counterfeit OOD probability $\rho$ is set to 15% and $\beta$ to 30.

| Threshold | Precision@1 | OOD F1 |
|---|---|---|
| 6 | 40.39 | 48.38 |
| 7 | 42.56 | 50.46 |
| 8 | 43.69 | 51.08 |
| 9 | 52.21 | 63.86 |
| 10 | 47.27 | 44.44 |

TABLE 7.6: Performances of AE-HCN-Indep on bAbI6 Test-OOD with different thresholds

| | bAbI6 Test | bAbI6 Test-OOD | |
|---|---|---|---|
| OOD Rate | Precision@1 | Precision@1 | OOD F1 |
| 5% | 55.25 | 55.48 | 69.72 |
| 10% | 55.08 | 57.29 | 74.73 |
| 15% | 55.04 | 55.35 | 70.38 |
| 20% | 53.48 | 56.53 | 75.55 |
| 25% | 53.72 | 56.66 | 73.13 |
| 30% | 54.87 | 56.02 | 71.44 |

TABLE 7.7: AE-HCN-CNN performance on bAbI6 with varying counterfeit OOD rates

Our training setup overall follows that in Experiment 1 (see Table 7.2). Although here, we pretrain the autoencoder on in-domain training data and keep it fixed while training the main models.

The result is shown in Table 7.5. Since there are multiple actions that are appropriate for a given dialogue context, we use per-utterance *Precision@K* as performance metric. We also report F1-score for the OOD detection to measure the balance between precision and recall. The performances of HCN on Test-OOD are about 15 points down on average from those on Test, showing the detrimental impact of OOD utterances to such models only trained on in-domain training data. AE-HCN(-CNN) outperforms HCN on Test-OOD by a large margin — about 17(20) points on average — while keeping the minimum performance trade-off compared to Test. Interestingly, AE-HCN-CNN has even better performance than HCN on Test, indicating that, with the CNN encoder, counterfeit OOD augmentation acts as an effective regularisation. In contrast, AE-HCN-Indep failed to robustly detect OOD utterances, resulting in much lower numbers for both metrics on Test-OOD as well as hurting the performance on Test. This result indicates two crucial points:

1. the inherent difficulty of finding an appropriate threshold value without actually seeing OOD data;

2. the limitation of the models which do not consider context.

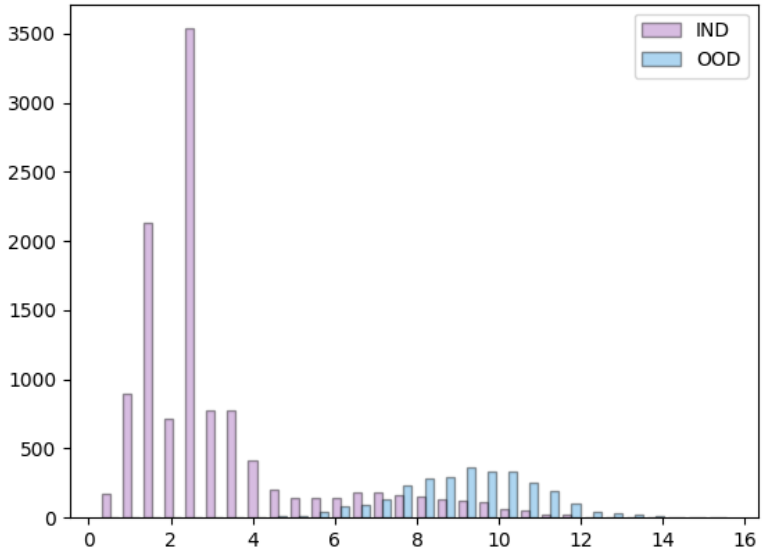



FIGURE 7.3: Histograms of AE reconstruction scores for the bAbI6 test data

For the first point, Figure 7.3 plots histograms of reconstruction scores for IND and OOD utterances of bAbI6 Test-OOD (the histograms for other datasets follow similar trends). If OOD utterances had been known a priori, the threshold should have been set to a much higher value than the maximum reconstruction score of IND training data (6.16 in this case).

For the second point, Table 7.6 shows the search for the best threshold value for AE-HCN-Indep on the bAbI6 task when given actual OOD utterances (which is highly unrealistic for the real-world scenario). Note that the best performance achieved at 9 is still not as good as that of AE-HCN(-CNN). This implies that we can perform better OOD detection by jointly considering other context features.

Furthermore, we conduct an experiment of AE-HCN's sensitivity to the $\beta$ hyperparameter which is the upper bound of the counterfeit reconstruction scores for turn dropout training. We vary $\beta$ from 30 which is quite close to $\alpha$ (the corresponding lower reconstruction score bound determined by the statistics of the trainset), up to 240 which is more that $10\alpha$. The results are shown in Table 7.8. As we observe in the table, there is no dependence between the overall accuracy of the model and $\beta$ parameter, however the OOD detection F1-score kept increasing as the range of reconstruction scores fed into the model was getting broader, and nearly stopped as $\beta$ neared $10\alpha$. Therefore, it makes sense to perform a grid search over $\beta$ while training the model.

Finally, we conduct a sensitivity analysis by varying counterfeit OOD probabilities. Table 7.7 shows performances of AE-HCN-CNN on bAbI6 Test-OOD with different $\rho$ values, ranging from 5% to 30%. The result indicates that our method manages to produce good performance

| $\alpha$ | $\beta$ | **Precision@1** | **OOD F1** |
|---|---|---|---|
| 23 | 30 | 59.3 | 75.3 |
| 23 | 60 | 57.8 | 75.5 |
| 23 | 90 | 56.9 | 75.6 |
| 23 | 120 | 59.3 | 75.9 |
| 23 | 150 | 58.3 | 76.1 |
| 23 | 180 | 58.0 | 76.2 |
| 23 | 210 | 57.0 | 75.7 |
| 23 | 240 | 59.1 | 76.0 |

TABLE 7.8: AE-HCN sensitivity to the $\beta$ hyperparameter on bAbI6 OOD-augmented testset

regardless of the $\rho$ value. This superior stability nicely contrasts with the high sensitivity of AE-HCN-Indep with regard to threshold values as shown in Table 7.6.

## 7.6 Conclusion

In this chapter, we explored the problem of robustness of neural dialogue systems to OOD input. Specifically, we presented a dataset for studying this problem along with a general procedure for augmenting arbitrary datasets of interest for such purpose. Secondly, we introduced turn dropout, a simple yet efficient technique for improving OOD robustness of dialogue control models and evaluated its effect on several Hybrid Code Network-family models.

We proposed a novel OOD detection method that does not require OOD data without any restrictions by utilising counterfeit OOD turns in the context of a dialogue. In the presence of OOD utterances, our method outperforms the best performing dialogue models to date equipped with an OOD detection mechanism by a large margin — more than 17 points in Precision@K on average — while minimising performance trade-off on in-domain test data. The detailed analysis sheds light on the difficulty of optimising context-independent OOD detection and justifies the necessity of context-aware OOD handling models.

This chapter concludes our contributions to data-efficiency of goal-oriented systems. In the next chapter, we are going to explore the area of social, chat-oriented dialogue as well as the data efficiency issues arising there. As such, in Chapters 4 and 5, while working on bootstrapping dialogue systems with minimum training data, we also cared about being able to train our systems without the need for any annotations. In the next chapter, we are going to pursue a similar goal in the chat-oriented dialogue area where the conversation models themselves or their key components are dependent on the annotations in the form of the user feedback scores.

# Chapter 8

# Data-Efficiency in Social Dialogue

In this final chapter, we are going to go beyond slot-filling goal-oriented conversations and look at open-domain social dialogue. Chat-oriented dialogue differs from goal-oriented in the sheer fact that there is no 'goal' to pursue. That makes the objectives of such interaction harder to define, and the conventional slot-value annotations widely used in goal-oriented dialogue become of little relevance in the chat-oriented setup. That is why large-scale conversational models (e.g. Vinyals and Le, 2015) were initially trained end-to-end from raw dialogue transcripts essentially to mimic the responses seen in the contexts they occur. However, in order to improve those models' base performance, especially in cases of user-faced products, annotations were brought back (e.g. Yu et al., 2016) — their simplest form being dialogue-level user ratings of the conversation, e.g. binary 'good/bad' or Likert-scale 1—5 scores.

As was discussed in Chapter 2, both generation and retrieval-based models were widely used for end-to-end chat-oriented dialogue. Under the prominent approach to such systems used in practical applications (such as Amazon Alexa Prize) — the *bot ensemble* (Serban et al., 2017a; Yu et al., 2016; Song et al., 2016) — a collection, or ensemble, of different bots is used, each of which proposes a candidate response to the user's input, and a *response ranker* selects the best response for the final system output to be uttered to the user. In this chapter, we focus on the task of finding the best supervision signal for training a response ranker for ensemble systems. Our contribution is twofold: first, we present a neural ranker[1] for ensemble-based dialogue systems and evaluate its level of performance using an annotation type which was provided to the Alexa Prize 2017 participants by Amazon (Ram et al., 2017), namely per-dialogue user ratings. Secondly and most importantly, we explore an alternative way of assessing social conversations simply via their *length*, thus removing the need for any user-provided ratings.

[1]Code and trained models available at http://tiny.cc/alana_ranker

## 8.1 Data Efficiency in Open-Domain Dialogue

Chatbots, or *socialbots*, are dialogue systems aimed at maintaining an open-domain conversation with the user spanning a wide range of topics, with the main objective of being engaging, entertaining, and natural. Currently, social chat systems are in great demand in industry both as a means to increase user retention with existing goal-oriented dialogue systems (e.g. Apple Siri, Google Assistant, or Amazon Alexa) and as standalone conversational systems for use e.g. in entertainment, robotics, and healthcare.

With the objective described above, it is problematic to build an open-domain, social dialogue system in a traditional, rule-based way (Weizenbaum, 1966) because hand-crafting such systems is not practical — they are difficult to create and maintain, cannot be trained on new data for a new setting, and cannot be automatically optimised. This is why many such systems are based on learning dialogue behaviour from massive amounts of data such as Reddit conversations (Al-Rfou et al., 2016; Liu et al., 2017), OpenSubtitles (Lison and Tiedemann, 2016) or the Ubuntu Dialog Corpus (Lowe et al., 2015).

There are two fundamental approaches to such systems. Generation-based models are predominantly based on SEQ2SEQ architecture (see Section 2.2.2) and produce responses word-by-word in a language model style given an encoded dialogue context representation. Ranking (or retrieval) based models work similarly to an Information Retrieval engine (see Section 2.2.1): given a (preprocessed) user utterance as input, they first collect a pool of candidate responses, and then use their own ranking function in order to select the best one. Common sources of candidate responses are normally search engines over conversational corpora (e.g. the above-mentioned Reddit conversations and OpenSubtitles), rule-based systems, question answering systems, or other response generation models. Ranking-based dialogue models are especially suitable for the ensembles of bots (Serban et al., 2017a; Yu et al., 2016; Song et al., 2016) which we are focusing on here.

### 8.1.1 The Need for Data Efficiency

It is well known that deep learning models are highly data-dependent, but there are currently no openly available data sources which can provide enough high-quality open-domain social dialogues for building a production-level socialbot. Therefore, a common way to get the necessary data is to collect it on a crowdsourcing platform (Krause et al., 2017). Based on the model type and the development stage, it may be necessary to collect either whole dialogues, or some form of human feedback on how good a particular dialogue or turn is. However, both kinds of data are time-consuming and expensive to collect.

The data efficiency of a dialogue model can be split into two parts accordingly:

| Variables | Pearson corr. coefficient |
|---|---|
| rating/length | 0.11 |
| rating/positive feedback | 0.11 |
| rating/negative feedback | 0.04 |
| length/positive feedback | 0.67 |
| length/negative feedback | 0.49 |

TABLE 8.1: Correlation study of key dialogue aspects

— *sample efficiency* — the number of data points needed for the model to train. As such, it is useful to specify an order of magnitude of the training set size for different types of machine learning models;

— *annotation efficiency* — the amount of annotation effort needed. For instance, traditional goal-oriented dialogue system architectures normally require intent, slot-value, and dialogue state annotation (e.g. Young et al., 2010), whereas end-to-end conversational models work simply with raw text transcriptions (e.g. Vinyals and Le, 2015).

### 8.1.2 Users' Ratings and Explicit Feedback

The 2017 Alexa Prize challenge made it possible to collect large numbers of dialogues between real users of Amazon Echo devices and various chatbots. The only annotation collected was per-dialogue ratings elicited at the end of the conversations by asking the user *"On a scale of 1 to 5, how much would you like to speak with this bot again"* (Venkatesh et al., 2017). Less than 50% of conversations were actually rated; the rest were quit without the user giving a score. In addition, note that a single rating is applied to an entire conversation (rather than individual turns), which may consist of very many utterances. The conversations in the challenge were about 2.5 minutes long on average, and about 10% of conversations were over 10 minutes long (Ram et al., 2017) — this makes the ratings very sparse. Finally, the ratings are noisy — some dialogues which are clearly bad can get good ratings from some users, and vice-versa (see a motivating example in Figure 8.1[2]).

Apart from users' ratings, we employed an additional metric of dialogue quality for our study — explicit user feedback. That is, we searched for dialogue turns containing positive or negative user's sentiment. Additionally, we used a whitelist and a blacklist of hand-picked phrases to filter out sentiments not addressed at the system directly. In total, we collected 605 unique utterances, e.g. *"that's pretty cool", "you're funny", "gee thanks", "interesting fact", "funny alexa you're funny"* and *"weird", "sounds dumb", "strange"*.

[2]Due to the restrictions on publicly presenting real conversational data from the challenge, we provide a sketch of a dialogue which we had with the system ourselves

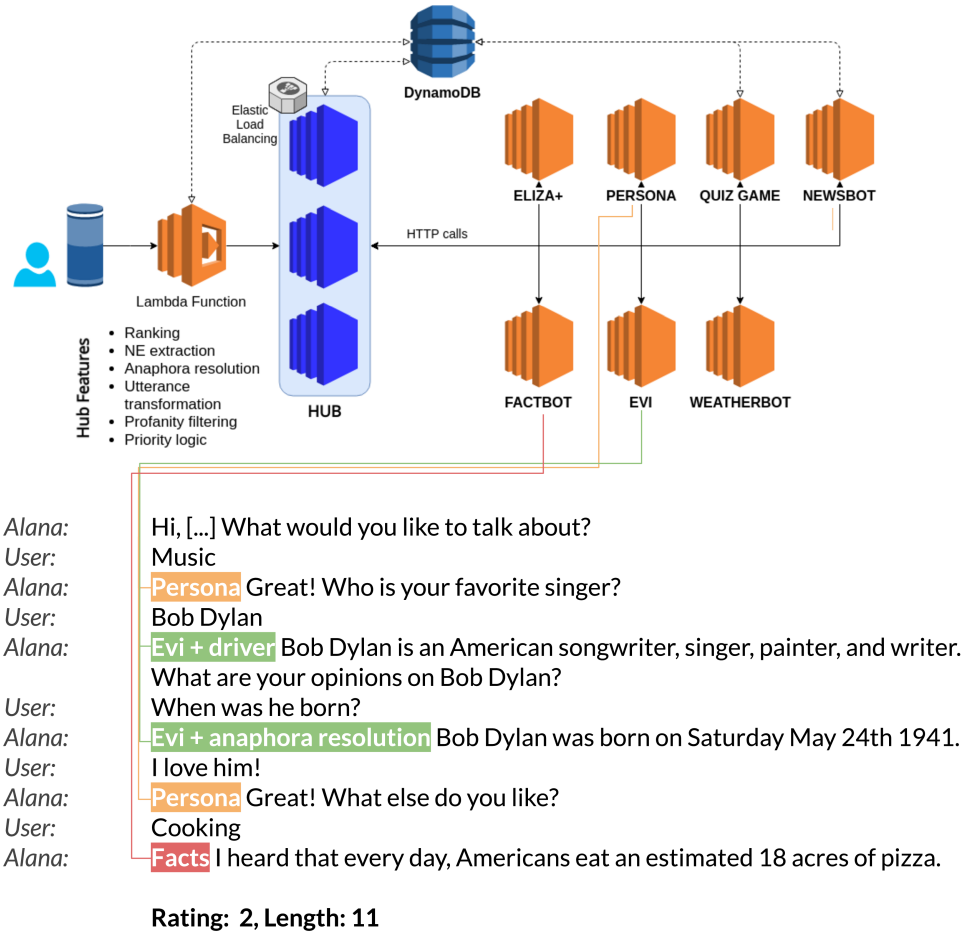



FIGURE 8.1: Alana architecture, with an example chat

Given the main objective of social dialogue stated in the Alexa Prize rules as 'long and engaging' conversation, we tried to verify an assumption that user ratings reflect these properties of the dialogue. Apart from our observations above, we performed a correlation analysis of user ratings and aspects of dialogue directly reflecting the objective: dialogue length and explicit user feedback (see Table 8.1). Although we have a significant number of dialogues which are both long and highly rated, the correlation analysis was not able to show any relationship between dialogue length and rating. Ratings aren't correlated with user feedback either (see Section 8.5 for the details of user feedback collection). On the other hand, we found a promising moderate correlation between the conversation length and explicit positive feedback from the users (we counted the number of such turns in a dialogue). The respective length/negative feedback relationship is slightly weaker.

Therefore, we experiment with conversation length for approximating user satisfaction and engagement and use it as an alternative measure of dialogue quality. This allows us to take advantage of all conversations, not just those rated by users, for training a ranker. While some

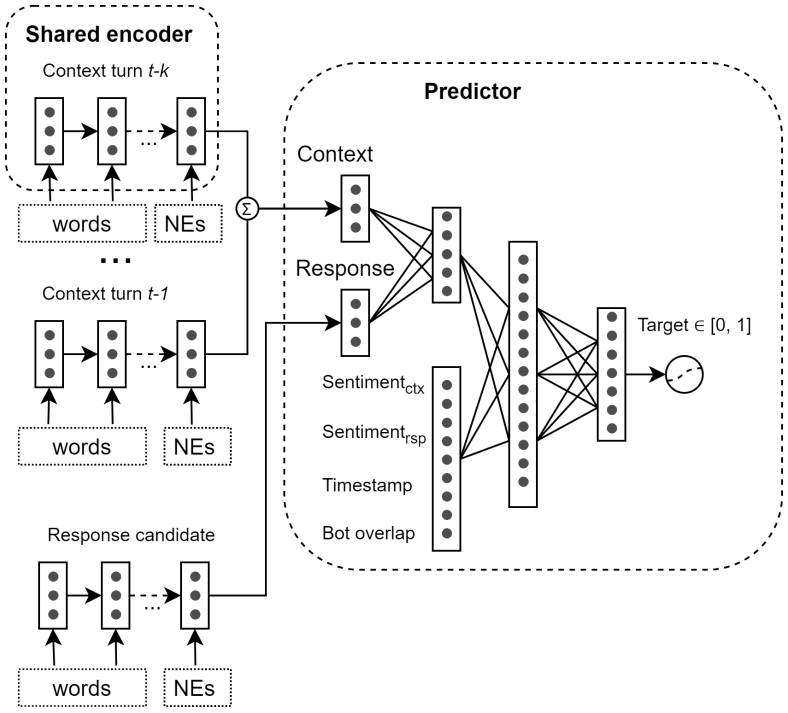



FIGURE 8.2: Neural ranker architecture

conversations might be long but not engaging (e.g. if there are a lot of misunderstandings, corrections, and speech recognition errors), training a ranker only using length makes it extremely *annotation-efficient*.

## 8.2 A Neural Ranker for Open-Domain Conversation

The ranker described here is part of Alana, Heriot-Watt University's Alexa Prize 2017 finalist socialbot (Papaioannou et al., 2017) — see the visualisation in Figure 8.1. Alana is an ensemble-based model incorporating information-retrieval-based bots with news content and information on a wide range of topics from Wikipedia, a question answering system, and rule-based bots for various purposes, from amusing users with fun facts to providing a consistent persona. The rule-based bots are also required to handle sensitive issues which can be raised by real users, such as medical, financial, and legal advice, as well as profanities.

### 8.2.1 Ranker Architecture

The architecture of our ranker[3] is shown in Figure 8.2. The inputs to the model are 1-hot word-by-word vectors of a candidate response and the current dialogue context (we use the 3 most recent system and user turns). They are encoded into a latent representation using a single shared RNN encoder based on GRU cells (Cho et al., 2014b). The context embedding vectors are then summed up and concatenated with the response embedding (Eq. 8.1):

$$\text{Enc}(C, r) = \sum_i \text{RNN}(C_i) \oplus \text{RNN}(r) \tag{8.1}$$

where $C$ is the dialogue context and $r$ is a response candidate.

The context and the response are represented using combined word-agent tokens (where agent is either a specific bot from the ensemble or the user) and are concatenated with the lists of named entities extracted using Stanford NER (Finkel et al., 2005). All the word-agent tokens and named entities share the same unified vocabulary.

Encoder outputs, along with additional dialogue features such as context and response sentiment, timestamp, and bot names in the context and the response, go into the *Predictor*, a feed-forward neural network (MLP) whose output is the resulting rating (Eq. 8.2):

$$\text{Pred}(C, r) = \sigma\left(L\left(\text{Sem}(C, r) \oplus f(C, r)\right)\right) \tag{8.2}$$

where: $L(x) = \text{ReLU}(Mx + b)$ is the layer used in the Predictor (the number of such layers is a model parameter),

$\text{Sem} = L\left(\text{Enc}(C, r)\right)$ is the vector of semantic context-response features, and

$f(C, r)$ is a vector of the additional dialogue features listed above.

We use ReLU activation for the hidden layers because it is known to be highly efficient with deep architectures (Nair and Hinton, 2010). Finally, we use sigmoid activation $\sigma$ for generating the final prediction in the range $[0, 1]$.

[3]We would like to point out that technically, our model produces a relevance *score* for a response candidate given the dialogue context, trained with a regression objective and only using training data with a partial order relation imposed (i.e. 1.0 for relevant, 0.0 for non-relevant context-response pairs). Therefore, the model can as well be considered a conversational response re-scorer. However, here we conform to the notation of Liu (2011) under which our approach is categorised as point-wise ranking model with the judgements in the form of relevance degrees (0/1).

### 8.2.2 Training Method

We use either dialogue rating or length as the prediction target scaled at $[0, 1]$ (as discussed in Sections 8.4 and 8.5). The model is trained to minimise the Mean Squared Error (MSE) loss against the target using the Adagrad optimiser (Duchi et al., 2011). In our training setup, the model learns to predict per-turn target values. However, since only per-dialogue ones are available in the data, we use the following approximation: the target value of a context-response pair is the target value of the dialogue containing it. The intuition behind this is an assumption that the majority of turns in "good" dialogues (either length- or rating-wise) are "good" in their local contexts as well — so that given a large number of dialogues, the most successful and unsuccessful turns will emerge from the corresponding dialogues.

## 8.3 Baselines

We compare our neural ranker to two other models also developed during the competition: handcrafted and linear rankers — all three were deployed live in the Alana Alexa Prize 2017 finalist system (Papaioannou et al., 2017), and were therefore of sufficient quality for a production system receiving thousands of calls per day. We also compare our model to a recently published dual-encoder response selection model by Lu et al. (2017) based on an approach conceptually close to ours.

### 8.3.1 Handcrafted Ranker

In the handcrafted approach, several turn-level and dialogue-level features are calculated, and a linear combination of those feature values with manually adjusted coefficients is used to predict the final ranking. The list of features includes:

— coherence, information flow, and dullness as defined by Li et al. (2016b);

— overlap between the context and the response with regards to named entities and noun phrases;

— topic divergence between the context turns and the response — topics are represented using the Latent Dirichlet Allocation (LDA, Hoffman et al., 2010);

— sentiment polarity, as computed by the NLTK Vader sentiment analyser (Gilbert and Hutto, 2014).[4]

[4] http://www.nltk.org/howto/sentiment.html

### 8.3.2 Linear Ranker

The linear ranker is based on the VowpalWabbit (VW) linear model (Agarwal et al., 2014). VW has a highly efficient implementation of stochastic gradient descent over various loss functions, capable of handling a very large (and sparse) feature space and a high number of training examples. We use the MSE loss function and the following features in our VW ranker model:

— bag-of-n-grams from the dialogue context (preceding 3 utterances) and the response,

— position-specific n-grams at the beginning of the context and the response (first 5 positions),

— dialogue flow features (Li et al., 2016b), the same as for the handcrafted ranker,

— bot name, from the set of bots in the ensemble.

VW implements feature combinations (Cartesian product) out-of-the-box, which allows it to naturally include a combination of n-gram features from the context and the response, as well as others — the detailed VW ranker configuration is shown in Table D.2 of Appendix D.

### 8.3.3 Dual-Encoder Ranker

The closest architecture to our neural ranker is that of Lu et al. (2017), who use a dual-encoder LSTM with a predictor MLP for task-oriented dialogue in closed domains. Unlike this work, they do not use named entities, sentiment, or other input features than basic word embeddings. Dialogue context is not modelled explicitly either, and is limited to a single user turn. We reproduced their architecture and set its parameters to the best ones reported in the original paper.

## 8.4 Training Data

Our data is transcripts of conversations between our socialbot and real users of the Amazon Echo collected over the challenge period, February – December 2017. The dataset consists of over 200,000 dialogues (5,000,000+ turns) from which over 100,000 dialogues (totalling nearly 3,000,000 turns) are annotated with ratings. From this data, we sampled two datasets of matching size for training our rankers, using the per-turn target value approximation described in Section 8.2.2 — the *Length* and *Rating* datasets for the respective versions of rankers.

The target values (length/rating) in both sets are normalised into the [0, 1] range, and the Length set contains context-response pairs from long dialogues (target value above 0.7) as positive

instances and context-response pairs from short dialogues (target value below 0.3) as negative ones. With the same selection criteria, the Rating set contains context-response pairs from highly rated dialogues (ratings 4 and 5) as the positive instances and context-response pairs from low-rated dialogues (ratings 1 and 2) as the negative ones. Each dataset contains 500,000 instances in total, with equal proportion of positive and negative instances. We use a 8:1:1 split for training, development, and test sets.

Prior to creating both datasets, we filtered out of the dialogue transcripts all system turns which cannot be treated as natural social interaction (e.g. a quiz game) as well as the outliers (interaction length $\geq$ 95th percentile or less than 3 turns long).[5] Thresholds of 0.3 and 0.7 were set heuristically based on preliminary data analysis. On the one hand, these values provide contrastive-enough ratings (e.g. we are not sure whether the rating in the middle of the scale can be interpreted as negative or positive). On the other hand, they allow us to get enough training data for both Length and Rating datasets.[6]

## 8.5 Evaluation and Experimental Setup

In order to tune the neural rankers, we performed a grid search over the shared encoder GRU layer size and the Predictor topology.[7] The best configurations are determined by the loss on the development sets. For evaluation, we used an independent dataset.

### 8.5.1 Evaluation Based on Explicit User Feedback

At the evaluation stage, we check how well the rankers can distinguish between the good responses and the bad ones. The criterion for 'goodness' that we use here is chosen to be independent from both training signals. Specifically, we collected an evaluation set composed of dialogue turns followed by explicit positive feedback from the users which was described earlier in Section 8.1.2. Our 'bad' response candidates are randomly sampled across the dataset. Please note: while we use ⟨positive feedback based response, random response⟩ pairs to evaluate the ranker, the opposite case, i.e. ⟨negative feedback based response, random response⟩ is not useful for the evaluation. The reason for that is, having 'gold' bad system responses (followed by the negative user feedback), it is not feasible to find the good alternatives to them in their exact contexts via random sampling. Due to the extremely high scarcity of the contexts, all the randomly sampled responses will most likely be bad.

[5] Some extremely long dialogues are due to users repeating themselves over and over, and so this filter removes these bad dialogues from the dataset. Dialogues less than 3 turns long are often where the user accidentally triggered the chatbot. These outliers amounted to about 14% of our data.

[6] Using more extreme thresholds did not produce enough data while less extreme ones did not provide adequate training signal.

[7] We tested GRU sizes of 64, 128, 256 and Predictor layers number/sizes of [128], [128, 64], [128, 32, 32].

| | | |
|---|---|---|
| **Context** | Usr | no |
| | Sys | i would love to talk about the news or maybe science, how about you? |
| | Usr | let's talk about the news |
| **Good response** | Sys | I read this somewhere - When will Hillary Clinton break her silence on Rick and Morty's Szechuan sauce? Shall I say some more about this? |
| **Bad response** | Sys | Hiya! What do you want to talk about? |
| **Context** | Usr | darth vader |
| | Sys | anakin skywalker the character in star wars episode iii revenge of the sith |
| | Usr | yeah that's him |
| **Good response** | Sys | Oh, regarding darth vader, I read this - Star Wars: Darth Vader 7 Reveals Another Jedi Who Survived the Clone Wars. Shall I say some more about this? |
| **Bad response** | Sys | Ok, I think you were talking about airplane, I heard that, An airplane or aeroplane is a powered, fixed-wing aircraft that is propelled forward by thrust from a jet engine or propeller. Do you want to know more about airplane? |
| **Context** | Usr | reading books same here |
| | Sys | oh a book worm how nice i like reading books on the cloud let's see what else |
| | Usr | harry potter harry potter |
| **Good response** | Sys | I believe that Harry Potter is a series of fantasy novels written by British author J. K. Rowling. |
| **Bad response** | Sys | So Are you more interested in sports or video games? |

Table 8.2: Examples from the User feedback dataset for pairwise ranking

'Goodness' defined in this way allows us to evaluate how well our two approximated training signals can optimise for the user's satisfaction as explicitly expressed at the turn level, thus leading to our desired behaviour, i.e., producing long and engaging dialogues. The *User feedback* dataset contains 24,982 ⟨ context, good response, bad response ⟩ tuples in total.

To evaluate the rankers on this dataset, we use *precision@k*, which is commonly used for information retrieval system evaluation (Eq. 8.3).

$$P@k(c, R) = \frac{\sum_{i=1}^{k} \text{Rel}\,(c, R_k)}{k} \tag{8.3}$$

where $c$ is dialogue context, $R$ is response candidates list, and Rel is a binary predicate indicating whether a particular response is relevant to the context.

Precision is typically used together with recall and F-measure. However, since our dialogue data is extremely sparse so that it is hard to find multiple good responses for the same exact dialogue context, recall and F-measure cannot be applied to this setting. Therefore, since we only perform pairwise ranking, we use *precision@1* to check that the good answer is the top-ranked one. Also due to data sparsity, we only perform this evaluation with *gold positive* responses and *sampled negative* ones — it is typically not possible to find a good response with exactly the same context as a given bad response.

| Model | P@1 (eval set) | Loss (test set) |
|---|---|---|
| Handcrafted | 0.478 | — |
| VowpalWabbit@length | 0.742 | 0.199 |
| VowpalWabbit@rating | 0.773 | 0.202 |
| DualEncoder@length | 0.365 | 0.239 |
| DualEncoder@rating | 0.584 | 0.247 |
| Neural@length | 0.824 | 0.139 |
| Neural@rating | **0.847** | **0.138** |

TABLE 8.3: Ranking models evaluation

### 8.5.2 Interim Results

The results of our first experiment — namely, pairwise ranking precision on the independent *User feedback* dataset and loss on the Length/Rating test sets (Section 8.4) for the corresponding trainset sizes of 500,000 — are shown in Table 8.3. We can see that the neural ranker trained with user ratings clearly outperforms all the alternative approaches in terms of test set loss on its respective dataset as well as pairwise ranking precision on the evaluation dataset. Also note that both versions of the neural ranker stand extremely close to each other on both evaluation criteria, given a much greater gap between them and their next-best-performing alternatives, the linear rankers.

The dual-encoder ranker turned out to be not an efficient model for our problem, partly because it was originally optimised for a different task as reported by Lu et al. (2017).

## 8.6 Training on Larger Amounts of Data

A major advantage of training on raw dialogue transcripts is data volume: in our case, we have roughly twice as many raw dialogues as rated ones (cf. Section 8.4). This situation is very common in data-driven development: since data annotation is a very expensive and slow procedure, almost always there is significantly more raw data than annotated data of a high quality. To illustrate this, we collected the extended training datasets of raw dialogues of up to 1,000,000 data points for training from the length signal. We trained our neural ranker and the VW ranker using the same configuration as in Section 8.5.

The results are shown in Figure 8.3, where we see that the neural ranker trained on the length signal consistently outperforms the ratings-based one. Its trend, although fluctuating, is more stable than that of VW — we believe that this is due to VW's inherent lower model capacity as well as its training setup, which is mainly optimised for speed. The figure also shows that VW@length is worse than VW@rating, regardless of training data size.

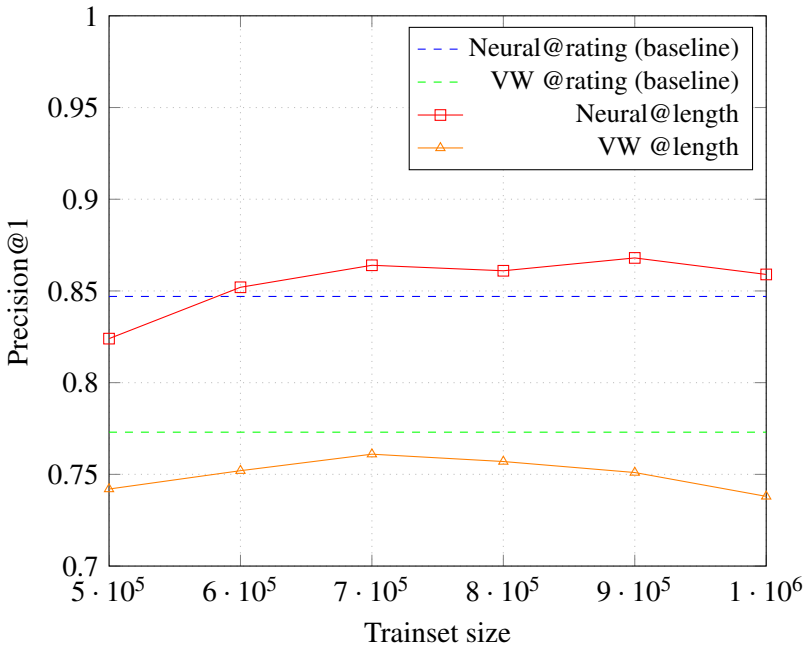



FIGURE 8.3: Comparison of rankers trained on extended datasets

## 8.7 Discussion

Our evaluation results show that the neural ranker presented above is an efficient approach to response ranking for social conversation. On a medium-sized training set, the two versions of the neural ranker, length and ratings-based, showed strongly superior performance to three alternative ranking approaches, and performed competitively with each other. Furthermore, the experiment with extended training sets shows that the accuracy of the length-based neural ranker grows steadily given more unannotated training data, outperforming the rating-based ranker with only slightly larger training sets.

The overall results of our experiments confirm that dialogue length, even approximated in quite a straightforward way, provides a sufficient supervision signal for training a ranker for a social conversation model.

## 8.8 Related Work

Work on response ranking for conversational systems has been been growing rapidly in recent years. Some authors employ ranking based on heuristically defined measures: Yu et al. (2015, 2016) use a heuristic based on keyword matching, part-of-speech filters, and WORD2VEC similarity. Krause et al. (2017) apply standard information retrieval metrics (TF-IDF) with importance weighting for named entities. However, most of the recent research attempts to train the ranking

function from large amounts of conversational data, as we do. Some authors use task-based conversations, such as IT forums (Lowe et al., 2015) or customer services (Lu et al., 2017; Kumar et al., 2018), while others focus on online conversations on social media (e.g. Wu et al., 2016; Al-Rfou et al., 2016).

The basic approach to learning the ranking function in most recent work is the same (e.g. Lowe et al., 2015; Al-Rfou et al., 2016; Wu et al., 2016): the predictor is taught to rank positive responses taken from real dialogue data higher than randomly sampled negative examples. Some of the approaches do not even include rich dialogue contexts and use only immediate context-response pairs for ranking (Ji et al., 2014; Yan et al., 2016; Lu et al., 2017). Some authors improve upon this basic scenario: Zhuang et al. (2018) take a desired emotion of the response into account; Liu et al. (2017) focus on the engagement of responses based on Reddit comments rating; Fedorenko et al. (2018) train the ranking model in several iterations, using highly ranked incorrect responses as negative examples for the next iteration. Nevertheless, to our knowledge, none of the prior works attempt to optimise for long-term dialogue quality; unlike in our work, their only ranking criterion is focused on the immediate response.

## 8.9 Conclusion

We have presented a neural response ranker for open-domain 'social' dialogue systems and described two methods for training it using the common supervision signals coming from conversational data: user-provided ratings and dialogue length. We demonstrated its efficiency by evaluating it using explicit positive feedback as a measure for user engagement. Specifically, trained on ratings, our neural ranker consistently outperforms several strong baselines; moreover, given larger amounts of data and only using conversation length as the objective, the ranker performs better the ratings-based one, reaching 0.87 Precision@1. This shows that conversation length can be used as an optimisation objective for generating engaging social dialogues, which means that we no longer need the expensive and time-consuming procedure of collecting per-dialogue user ratings, as was done for example in the Alexa Prize 2017 and is common practice in conversational AI research. Per-turn user ratings may still be valuable to collect for such systems, but these are even more expensive and problematic to obtain. Looking ahead, this advance will make data collection for social conversational agents simpler and less expensive in the future.

# Chapter 9

# Conclusions and Future Work

In this thesis, we have presented a series of techniques towards enabling the development of data-efficient and robust dialogue systems in a data-driven way.

The core contributions of this thesis are the models for bootstrapping dialogue systems from minimal data. Our dialogue knowledge transfer model DIKTNET addresses the problem of training dialogue response generation systems in a few-shot setup, and the hybrid GRTR model is designed for the adaptation to a new domain where there exists support data to retrieve responses from.

Our subsequent contributions presented in this thesis address potentially insufficient robustness of the models trained from minimal data, addressing spoken disfluency detection and OOD input detection problems. The multitask LSTM-based disfluency detector supports incremental word-by-word processing and demonstrates generalisation potential beyond its main dataset — therefore, it can be used with a wide range of dialogue models, potentially improving their coverage of naturally varied input data with no extra training effort. In turn, the data-augmentation based training procedure for OOD handling is potentially applicable to the setups with the strictest training data limitations as it does not require having real OOD examples to adjust to and estimates the odds of encountering an unusual utterance in usual contexts.

Finally, our study on data efficiency in social dialogue sheds light on a way to continuously improve the performance of open-domain response rankers only using dialogue length as the main supervision signal — thus avoiding the dependence on user ratings which are cumbersome and expensive to collect and often times are overly noisy for the direct usage as a supervision signal.

## 9.1 Directions for Future Work

The findings we got from our studies in this thesis open up possibilities for further research — here, we will briefly outline the most promising directions of future work.

Firstly, the DIKTNET presented in Chapter 4, while outperforming the previous best model on the Stanford Multi-Domain dataset in both accuracy and the amount of in-domain data required, still has got several areas of improvement. We see a potential benefit in bringing more structure to the latent dialogue representation which would intuitively correspond better to the actual structure of dialogue, e.g. as shown in Shi et al. (2019b). Another potential point of improvement is the system's goal-oriented performance: we assume that in a few-shot setup, it highly depends on the way in which the system handles the KB entries: a study of the particular copy mechanism employed in our setup and its optimality for the task might shed light on the ways to attain higher Entity F1 scores. Finally, an evaluation beyond the word overlap-based metrics is necessary for the adequate assessment of the system's performance in real-world settings.

Similarly, the absolute results of GRTR, the winning entry at Dialog System technology Challenge 8 Fast Domain Adaptation task which we presented in Chapter 5, still suggest that domain adaptation for response generation needs further research towards achieving production-level performance. One promising direction that we are going to explore in our future work is the meta-learning framework, e.g. as used by Qian and Yu (2019). Based on splitting the task into multiple subtasks — with an independent copy of the base model for each one — and subsequently incorporating the individual training progress back into the base model, meta-learning approach will naturally fit the multi-domain, multi-task nature of the MetaLWOz dataset as well as lead to a potentially better fine-tuning performance.

For the multitask LSTM disfluency detector presented in Chapter 6, we see the main issue being the modest out-of-dataset generalisation performance. Although it was a common issue for all the models we evaluated, attaining a more practical level of generalisation is key for making this model a truly reusable component for a wide range of dialogue system pipelines — therefore, addressing this issue is our next step in this direction. As such, we will explore possibilities of knowledge transfer to new closed domains in a 1-shot setting, both with regular supervised training and unsupervised LM fine-tuning.

Our study on robustness to out-of-domain input presented in Chapter 7 of this thesis also leaves space for further exploration. As such, we presented and evaluated a series of models for the detection of OOD utterances — all based on autoencoders of different types. It therefore makes sense to explore other ways of scoring OOD utterances than autoencoders: for example, Generative Adversarial Networks (GANs) have great potential, especially for their inherent capabilities to produce realistic data samples. We are also interested in using traditional generative models to produce more realistic counterfeit user utterances. Finally, it is worth noting that our

method is conceptually designed for improving robustness in extreme data-efficient settings (i.e. 1-shot/few-shot training) by not requiring any other data than the available IND dialogue examples. However, in the interest of comparability with the original HCN model, we performed our experiments with full data — therefore, few-shot evaluation of our models is the immediate next step in this research direction for us.

The final study of this thesis presented in Chapter 8 addressed the problems of social, or chat-oriented dialogue. As such, our neural model for conversational response ranking together with the technique of training it from raw data confirmed that dialogue length, even approximated in quite a straightforward way, provides a sufficient supervision signal for training a ranker for a social conversation model. In our future work, we will attempt to further improve the model using the same data in an adversarial setup following Wang et al. (2017a). We also plan to directly train our model for pairwise ranking in the fashion of Burges et al. (2005) instead of the current point-wise approach. Finally, we are going to employ contextual sampling of negative responses using the approximate nearest neighbour search (Johnson et al., 2017) in order to perform a more efficient pairwise training.

The above techniques have the potential to make an impact in industrial data-driven dialogue system development. For that, they can be incorporated into a user-friendly 'data efficiency toolkit' aimed at usage by non-experts. As the main way to control the behaviour of the purely data-driven systems assumes passing the corresponding training examples into the pipeline, there is the need for the means to analyse and monitor the performance of the dialogue system's key components in a transparent way for the end user. This can be done in terms of the specific decisions made by the system and the training examples contributed to those. Although explainable machine learning was not the scope of this work, it is important to point out that such means are vital for a data-driven product targeted at non-expert users.

Crucially, the models and techniques presented here were evaluated in certain setups which are mainly motivated by reproducibility and fair comparison to the previous models for the respective tasks. These setups do not necessarily correspond to the designated settings of limited training data. For example, the disfluency detection model as well as the OOD detection models were trained and evaluated on the full datasets, and therefore a few-shot evaluation with the possible subsequent fine-tuning of the models is needed in order to make the them practically justified.

We see a strong potential for a significant impact — both in academia and industry — of the data-efficient techniques presented in this thesis and will continue pursuing efforts to ensure wide applicability of those in real-world scenarios following the steps outlined above. We hope that the open-source resources released as part of this work will foster further research in this direction across the dialogue research community.

# Appendix A

# Dialogue Knowledge Transfer Networks — Supplementary Material

| | Navigation | | Weather | | Schedule | |
|---|---|---|---|---|---|---|
| | BLEU, % | Entity F1, % | BLEU, % | Entity F1, % | BLEU, % | Entity F1, % |
| ZSDG | 5.9 | 14.0 | 8.1 | 31 | 7.9 | 36.9 |
| NLU_ZSDG | 6.1 ± 2.2 | 12.7 ± 3.3 | 5.0 ± 1.6 | 16.8 ± 6.7 | 6.0 ± 1.7 | 26.5 ± 5.4 |
| NLU_ZSDG+Stage1 | 7.9 ± 1 | 12.3 ± 2.9 | 8.7 ± 0.6 | 21.5 ± 6.2 | 8.3 ± 1 | 20.7 ± 4.8 |
| HRED@1% | 6.0 ± 1.8 | 9.8 ± 4.8 | 6.9 ± 1.1 | 22.2 ± 10.7 | 5.5 ± 0.8 | 25.6 ± 8.2 |
| HRED@3% | 7.9 ± 0.7 | 11.8 ± 4.4 | 9.6 ± 1.8 | 39.8 ± 7 | 8.2 ± 1.1 | 34.8 ± 4.4 |
| HRED@5% | 8.3 ± 1.3 | 15.3 ± 6.3 | 11.5 ± 1.6 | 38.0 ± 10.5 | 9.7 ± 1.4 | 37.6 ± 8.0 |
| HRED@10% | 9.8 ± 0.8 | 19.2 ± 3.2 | 12.9 ± 2.4 | 40.4 ± 11.0 | 12.0 ± 1.0 | 38.2 ± 4.2 |
| HRED+VAE@1% | 3.6 ± 2.6 | 9.3 ± 4.1 | 6.8 ± 1.3 | 23.2 ± 10.1 | 4.6 ± 1.6 | 28.9 ± 7.3 |
| HRED+VAE@3% | 6.9 ± 1.9 | 15.6 ± 5.8 | 9.5 ± 2.6 | 32.2 ± 11.8 | 6.6 ± 1.7 | 34.8 ± 7.7 |
| HRED+VAE@5% | 7.8 ± 1.9 | 12.7 ± 4.2 | 10.1 ± 2.1 | 40.3 ± 10.4 | 8.2 ± 1.7 | 34.2 ± 8.7 |
| HRED+VAE@10% | 9.0 ± 2.0 | 18.0 ± 5.8 | 12.9 ± 2.2 | 40.1 ± 7.6 | 11.6 ± 1.5 | 39.9 ± 6.9 |
| HRED+Stage1@1% | 7.1 ± 0.8 | 10.1 ± 4.5 | 10.6 ± 2.1 | 31.4 ± 8.1 | 7.4 ± 1.2 | 29.1 ± 6.6 |
| HRED+Stage1@3% | 9.2 ± 0.8 | 14.5 ± 4.8 | 13.1 ± 1.7 | 40.8 ± 6.1 | 9.2 ± 1.2 | 32.7 ± 6.1 |
| HRED+Stage1@5% | 10.3 ± 1.2 | 15.6 ± 4.5 | 14.5 ± 2.2 | 40.9 ± 8.6 | 11.8 ± 1.9 | 37.6 ± 6.1 |
| HRED+Stage1@10% | 12.3 ± 0.9 | 17.3 ± 4.5 | 17.6 ± 1.9 | 47.5 ± 6.0 | 15.2 ± 1.6 | 38.7 ± 8.4 |
| HRED+ELMo@1% | 5.8 ± 1.9 | 18.2 ± 3.8$^\star$ | 7.3 ± 2.6 | 38.5 ± 11.1 | 6.3 ± 2.6 | 36.3 ± 9.2 |
| HRED+ELMo@3% | 8.0 ± 1.3 | 17.2 ± 4.2 | 10.6 ± 1.1 | 42.0 ± 11.0 | 9.5 ± 2.0 | 39.6 ± 9.2 |
| HRED+ELMo@5% | 9.4 ± 0.8 | 21.5 ± 7.3 | 12.1 ± 2.0 | 39.0 ± 12.8 | 11.3 ± 2.1 | 40.0 ± 5.6 |
| HRED+ELMo@10% | 9.9 ± 1.1 | 24.3 ± 5.7 | 14.9 ± 2.7 | 41.4 ± 12.0 | 14.5 ± 1.4 | 43.4 ± 3.9 |
| **DiKTNet@1%** | **8.4 ± 0.7$^\star$** | **15.2 ± 4.0** | **11.5 ± 1.7$^\star$** | **43.0 ± 10.5$^\star$** | **8.1 ± 0.8$^\star$** | **40.5 ± 6.3$^\star$** |
| DiKTNet@3% | 10.4 ± 1.2 | 19.2 ± 4.8 | 15.7 ± 2.1 | 44.0 ± 11.7 | 11.1 ± 1.3 | 38.2 ± 5.8 |
| DiKTNet@5% | 11.5 ± 1.1 | 23.9 ± 2.9 | 15.5 ± 2.1 | 39.5 ± 14.8 | 13.7 ± 2.0 | 41.1 ± 3.8 |
| DiKTNet@10% | 12.9 ± 1.0 | 26.8 ± 4.2 | 20.4 ± 1.2 | 48.0 ± 5.6 | 17.5 ± 1.3 | 42.8 ± 2.6 |

Table A.1: Evaluation results. Marked with asterisks are individual results higher than ZSDG's performance and which are achieved with the minimum amount of training data. In bold is the model consistently outperforming ZSDG in all domains and metrics with minimum data.

| Domain | #Dialogues | Domain | #Dialogues |
|---|---|---|---|
| UPDATE CALENDAR | 1991 | GUINESS CHECK | 1886 |
| ALARM SET | 1681 | SCAM LOOKUP | 1658 |
| PLAY TIMES | 1601 | GAME RULES | 1590 |
| CONTACT MANAGER | 1483 | LIBRARY REQUEST | 1339 |
| INSURANCE | 1299 | HOME BOT | 1210 |
| HOW TO BASIC | 1086 | CITY INFO | 965 |
| TIME ZONE | 951 | TOURISM | 935 |
| SHOPPING | 903 | BUS SCHEDULE BOT | 898 |
| CHECK STATUS | 784 | WHAT IS IT | 776 |
| STORE DETAILS | 737 | APPOINTMENT REMINDER | 668 |
| PRESENT IDEAS | 664 | GEOGRAPHY | 653 |
| SKI BOT | 607 | MOVIE LISTINGS | 607 |
| UPDATE CONTACT | 581 | ORDER PIZZA | 577 |
| EDIT PLAYLIST | 574 | SPORTS INFO | 561 |
| BOOKING FLIGHT | 555 | WEATHER CHECK | 551 |
| EVENT RESERVE | 539 | RESTAURANT PICKER | 535 |
| LOOK UP INFO | 533 | AUTO SORT | 514 |
| QUOTE OF THE DAY BOT | 513 | WEDDING PLANNER | 510 |
| MAKE REST. RESERVATIONS | 510 | AGREEMENT BOT | 507 |
| NAME SUGGESTER | 499 | APARTMENT FINDER | 499 |
| HOTEL RESERVE | 497 | PHONE PLAN BOT | 496 |
| DECIDER BOT | 487 | VACATION IDEAS | 486 |
| PHONE SETTINGS | 473 | POLICY BOT | 447 |
| PROMPT GENERATOR | 446 | MUSIC SUGGESTER | 445 |
| PET ADVICE | 426 | BANK BOT | 367 |
| CATALOGUE BOT | 288 | | |

TABLE A.2: METALWOZ domains

## METALWOZ — dialogue examples

**Domain**: Ski

**User Task**: You want to know if there are good ski hills an hour's drive from your current location.

**Bot Task**: Tell the user that there are no ski hills in their immediate location.

Bot Hello how may I help you?

User Is there any good ski hills an hour's drive from my current location?

Bot I'm sorry to inform you that there are no ski hills in your immediate location

User Can you help me find the nearest?

Bot Absolutely! It looks like you're about 3 hours away from Bear Mountain. That seems to be the closest.

User Hmm.. sounds good

Bot Alright! I can help you get your lift tickets now!When will you be going?

User Awesome! please get me a ticket for 10pax

Bot You've got it. Anything else I can help you with?
User None. Thanks again!
Bot No problem!

**Domain**: Game rules
**User Task**: Ask how to win at the text adventure Zork. If the Bot cannot help you, ask them what games it can help you with.
**Bot Task**: Tell the user that you are programmed to help them play games, not win them. If the user starts talking about other games, go along with them and try to help.
Bot Hello how may I help you?
User I want to know how I can win playing Zork?
Bot I am programmed to help you play games, not win them
User What games can you help me with?
Bot Any game
User Can you specify a certain game?
Bot Zork
User What are the rules of Zork?
Bot You have to follow through a cave adventure
User What is the ending of the game?
Bot The user comes out of the cave

**Domain**: Music suggestion
**User Task**: Tell the bot that you like a certain genre of music, but don't prompt it to give you suggestions
**Bot Task**: Provide suggestions, even if the user doesn't ask for them (or tells you they don't need any)
Bot Hello how may I help you?
User Hello. I like classic rock music.
Bot Dream on Aerosmith
User No, I'm not a huge fan of that song. Is there anything else available?
Bot Sweet Home Alabama
User Well, I do like Lynyrd Skynyrd, but that song is so cliche. Do you have anything else by the same band?
Bot Free bird
User Yes, that is the perfect song for how I am feeling right now!
Bot Great, Is there anything else?
User No, but I really appreciate your assistance. Thank you so much!
Bot Youre welcome

**Domain**: Information retrieval

**User Task**: Describe a movie where these guys have to climb a mountain and then throw a thing into the mountain and ask the bot if they know what the movie is called.

**Bot Task**: Ask the user to elaborate as much as possible about the movie they are asking about, and then suggest they're thinking of Lord of the Rings.

Bot Hello how may I help you?

User I'm trying to remember this movie. Can you help?

Bot I would be happy to help. Can you tell me what it is about?

User In it these guys have to climb up a mountain. Then they throw something into the mountain.

Bot Do you know what genre the movie is, such as comedy, horror, sci-fi, or fantasy?

User Not sure about that.

Bot Do you any of the actor or character names in the movie?

User No... All I can remember is the stuff about the mountain.

Bot Is it a new movie, or an older movie?

User I don't know.

Bot It sounds like the movie you are talking about could be Lord of the Rings. Does that sound right?

User It could be that. Thanks for the help.

# Appendix B

# Generative-Retrieval Transformers — Supplementary Material

| | BLEU-1 | BLEU-2 | BLEU-3 | CIDEr | METEOR | ROUGE-L |
|---|---|---|---|---|---|---|
| Retrieval, BERT | 7.93 | 4.43 | 2.87 | 12.56 | 7.38 | 6.91 |
| Retrieval, SP+FT | 9.57 | 5.37 | 3.45 | 14.32 | 6.98 | 7.19 |
| HRED | 8.66 | 3.86 | 2.11 | 13.73 | 6.02 | 7.75 |
| GPT-2 base[1] | 8.2 | 3.95 | 2.22 | 16.41 | 6.1 | 8.34 |
| GPT-2 +sup[2] | 11.33 | 6.45 | 4.17 | 23.38 | 8.23 | 10.74 |
| **GRTR** | **12.73** | **7.43** | **4.88** | **28.74** | **9.18** | **11.77** |

[1] does not use support set.[2] fine-tuned to support set, but does not use retrieval logic

TABLE B.1: Automatic evaluation results on METALWOZ pure task

| | BLEU-1 | BLEU-2 | BLEU-3 | CIDEr | METEOR | ROUGE-L |
|---|---|---|---|---|---|---|
| Retrieval, BERT | 5.35 | 2.16 | 1.05 | 4.98 | 4.56 | 4.52 |
| Retrieval, SP+FT | 5.94 | 2.25 | 0.93 | 4.69 | 4.29 | 4.53 |
| HRED | 8.94 | 3.87 | 2.02 | 12.65 | 6.05 | 7.55 |
| GPT-2 base | 8.37 | 3.8 | 2.05 | 15.6 | 6.17 | 8.55 |
| GPT-2 +sup | 10.21 | 5.26 | 2.95 | 18.06 | 7.06 | **9.59** |
| **GRTR** | **10.39** | **5.31** | **2.95** | **18.26** | **7.1** | 9.27 |

TABLE B.2: Automatic evaluation results on METALWOZ cross-task

| | | |
|---|---|---|
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I'm trying to book rooms |
| | Wiz | For where? |
| | Usr | I need a few hotel rooms in Tucson |
| | Wiz | how many total rooms? |
| **Gold response** | Usr | I need ten rooms |
| **Generated candidate** | Usr | Five rooms [**2.794**] |
| **Retrieved candidate** | Usr | I need 4 rooms on the same floor [2.793] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | Want some info about Cyprus. |
| | Wiz | What would you like to know about Cyprus? |
| **Gold response** | Usr | What's best time to visit there? |
| **Generated candidate** | Usr | What is the best time to visit? [**1.088**] |
| **Retrieved candidate** | Usr | What is the best time to visit Cyprus? [1.087] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I need some help. |
| | Wiz | yes, how can i help you? |
| | Usr | I'm going to Montreal, and good festivals? |
| | Wiz | yes, there will have music festivals and dancing party. and wine testing party. |
| | Usr | In the summer? |
| | Wiz | yes, it will in the august. |
| | Usr | September? |
| | Wiz | there only have wine testing party for september. |
| **Gold response** | Usr | Bummer |
| **Generated candidate** | Usr | ok thanks [-1.282] |
| **Retrieved candidate** | Usr | That sounds interesting. Can you book me for that day? [**−1.274**] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I need some ideas |
| | Wiz | Sure About what? |
| | Usr | Where's a good place to bring kids for vacation |
| | Wiz | I highly suggest Disney Land |
| | Usr | That's a good idea. When is a good time for us to go? When it's not so hot? |
| | Wiz | But the place is vey expensive |
| **Gold response** | Usr | Yeah, I'm okay with that |
| **Generated candidate** | Usr | I don't think that's a good idea. What are the best places to bring kids? [0.375] |
| **Retrieved candidate** | Usr | Oh I know we'd never go if we didn't get a Florida discount on tickets [**0.384**] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I need to book a flight and i need it by tonight |
| | Wiz | Ok, book a flight, it will be expensive at this late date |
| | Usr | Book me a flight to hawaii for 2 people leaving tonight |
| | Wiz | Ok, let me see what I have, I have two tickets for 100,000 dollars |
| **Gold response** | Usr | Book it. Money is no limit |
| **Generated candidate** | Usr | Money is not a problem for us [0.805] |
| **Retrieved candidate** | Usr | Sounds good I'll take them [**0.819**] |

TABLE B.3: GRTR predictions with the closest generated/retrieved candidates — METALWOZ pure task (in bold is the model's final response)

| | | |
|---|---|---|
| **Context** | Wiz | Hello how may I help you? |
| | Usr | Hello. I need to book a flight for two |
| | Wiz | ok where are you going? |
| | Usr | I will be heading to Hawaii and I need to leave tonight |
| | Wiz | will you be bringing extra luggage? |
| | Usr | No, I will need first class seats. Money is no object |
| | Wiz | i have 2 tickets for $50,000 |
| | Usr | Okay, that will be fine. Please purchase with my credit card on file |
| | Wiz | just to verify, what is the last 2 digits of your credit card? |
| **Gold response** | Usr | The last two digits of my credit card are 00 |
| **Generated candidate** | Usr | The last 2 digits are: 5555 [**5.081**] |
| **Retrieved candidate** | Usr | Sure. Just sent it to my email. Thanks! [−9.761] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I need an idea. |
| | Wiz | Sure, what's up? |
| | Usr | Can use advise me on a vacation spot? |
| | Wiz | Yes. Disney World and Disney Land are always fun for children and adults, but it's really expensive |
| | Usr | Anything closer? |
| | Wiz | Have you ever been to Niagara Falls? |
| | Usr | Yes,but that's too far away. |
| | Wiz | Where is your location? |
| **Gold response** | Usr | Nebraska |
| **Generated candidate** | Usr | I'm in New York. [**3.711**] |
| **Retrieved candidate** | Usr | ok, thanks [−9.605] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I need to book a room for me |
| | Wiz | OK. Where do you want to stay? |
| | Usr | In downtown Toronto |
| | Wiz | And for what dates would you like the room? |
| | Usr | I need a 2 bedroom on the second floor I'd like to stay for this weekend |
| | Wiz | And how many people will be in your party? |
| **Gold response** | Usr | 2 |
| **Generated candidate** | Usr | Just myself and a few friends please [**3.82**] |
| **Retrieved candidate** | Usr | Price isn't a concern. I would like a room on the second floor with 2 beds if that helps. [−9.183] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I need some ideas. |
| | Wiz | Okay, what about? |
| | Usr | I'm going to be taking my vacation soon |
| | Wiz | Where will you like to go? |
| **Gold response** | Usr | I need someone good for the kids. somewhere* |
| **Generated candidate** | Usr | Well, I'm going to be in Miami [**2.13**] |
| **Retrieved candidate** | Usr | yes please. [−10.672] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I have 5 days of vacation next month I would like to go somewhere |
| | Wiz | Do you have any favorite places you have in mind? |
| | Usr | Anywhere in colorado, wyoming, montana |
| | Wiz | Let me look that up for you! If you're planning for Denver, Colorado. I see a few popular tourist attractions: Denver Art Museum and Elitch Gardens Theme Park Would you like to vacation there, or would you like me to continue based on your list of favorite places? |
| **Gold response** | Usr | That's a good idea. When is a good time for us to go? When it's not so hot? |
| **Generated candidate** | Usr | Sure, I'll take it! Thanks for the suggestions [**-0.11**] |
| **Retrieved candidate** | Usr | What are some cheap Florida places to go? [−11.762] |

TABLE B.4: GRTR predictions with the most distant generated/retrieved candidates — METALWOZ pure task (in bold is the model's final response)

| | | |
|---|---|---|
| **Context** | Wiz | Hello how may I help you? |
| | Usr | help me book a flight |
| | Wiz | do you have a particular place in mind? |
| | Usr | i hear Greece is nice this time of year. how do i get a flight there? |
| | Wiz | yes, it is; i can book your flights now, if you like |
| | *Usr* | wait don't! i am only curious on how to fly there |
| | Wiz | AA Airlines fly to Greece everyday |
| | *Usr* | how much is the ticket? |
| | *Wiz* | we have specials this month for $350 roundtrip |
| **Gold response** | Usr | i don't have enough funds. i'll save up |
| **Generated candidate** | Usr | how much is the flight to Germany? [**1.569**] |
| **Retrieved candidate** | Usr | thanks [1.57] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | Hello bot! |
| | Wiz | Hi, how can I help you? |
| | Usr | Are there any good festivals in Montreal this summer? |
| | Wiz | Yes there are a few, would you like me to list them? |
| | *Usr* | List the best one. |
| | Wiz | In my opinion the best one is the beer festival! |
| **Gold response** | Usr | Awesome, I love to drink. How much are tickets. |
| **Generated candidate** | Usr | Okay. I'll go with that. [**1.754**] |
| **Retrieved candidate** | Usr | I like that idea. [1.752] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | Hi! I need help with my travelling. |
| | Wiz | How can I help you? Okay good. I can offer you a flight to Franc, England, Germany, and Italy |
| | Usr | Sure. Can you get me a flight to Germany by next tuesday? |
| | Wiz | Alright done. Flight to Gemany. |
| **Gold response** | Usr | How much does it cost? |
| **Generated candidate** | Usr | Okay. I need to book a flight to Germany on Monday. [−**1.916**] |
| **Retrieved candidate** | Usr | Perfect, what's the price tag? [−1.921] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I am heading to Montreal this Summer, any good festivals around that time? |
| | Wiz | Montreal Pride |
| | Usr | What will I do there? |
| | Wiz | It's a music fest |
| **Gold response** | Usr | Great! I like that bot |
| **Generated candidate** | Usr | Is there anything else? [−0.309] |
| **Retrieved candidate** | Usr | Other suggestions? [−**0.3**] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | i need help with vacations |
| | Wiz | How can I help you with that? |
| | Usr | whats a good place to bring kids |
| | Wiz | Disney World or Disney Land is always fun! A little pricey but fun. |
| **Gold response** | Usr | anything else? |
| **Generated candidate** | Usr | ok i will go there [0.799] |
| **Retrieved candidate** | Usr | sounds good, book tickets [**0.812**] |

TABLE B.5: GRTR predictions with the closest generated/retrieved candidates — METALWOZ cross-task (in bold is the model's final response)

| | | |
|---|---|---|
| **Context** | Wiz | Hello how may I help you? |
| | Usr | i need a room |
| | Wiz | Sure, where are you looking for a room at? |
| | Usr | with 2 bedrooms |
| | Wiz | Okay, I can help with that. |
| | Usr | and it is located on the second floor of a hotel |
| | Wiz | Okay, what city are you interested in? |
| | Usr | te specific place is in toronto |
| | Wiz | What date would you like to book the room for? |
| **Gold response** | Usr | Next monday |
| **Generated candidate** | Usr | May 16th [**2**.**642**] |
| **Retrieved candidate** | Usr | Thanks a lot! [−13.151] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | Hey, I've got a question about tourism |
| | Wiz | Sure, I can help with that. |
| | Usr | I'm curious if there are any bars around here |
| | Wiz | Could you specify your location for me? |
| **Gold response** | Usr | Where I am? |
| **Generated candidate** | Usr | I'm in Tampa, I need some info on the best time to visit the city [**3**.**008**] |
| **Retrieved candidate** | Usr | I need some information about Rome and where to go. [−12.744] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I'm here visiting, are there any good bars in the area? |
| | Wiz | I'm sorry I can't help unless you clarify your question |
| | Usr | I'm looking for good bars in the area, craft beer or wine? |
| | Wiz | What area? |
| | Usr | Downtown in the city |
| | Wiz | What city? |
| **Gold response** | Usr | Tampa |
| **Generated candidate** | Usr | Montreal [**4.48**] |
| **Retrieved candidate** | Usr | Ok thanks for the tip [−10.164] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I am a history buff. Do you know if there are any great historical sites to visit in Rome? |
| | Wiz | Leonardo da Vinci was one of the greatest inventors of all time and they have a museum for him there |
| | Usr | Oh that is so cool. I want to see that. Are there any other places to visit? |
| | Wiz | Well if you want to go look at a bunch of old bones of dead people that have died throughout the ages you can always go to the catacombs |
| **Gold response** | Usr | Okay that sounds like a great place to go also. Keep the ideas coming |
| **Generated candidate** | Usr | Oh, that sounds fun. I'll do that. [**2**.**109**] |
| **Retrieved candidate** | Usr | I would like to know more about the air-balloon rides [−12.116] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | Can you help me with information on Montreal? |
| | Wiz | What would you like to know? |
| **Gold response** | Usr | Are there any festivals there in the summer? |
| **Generated candidate** | Usr | What is the tallest building there? [**1**.**905**] |
| **Retrieved candidate** | Usr | Ok, thank you. I was wondering where in Rome I can visit to learn about it's history. [−12.161] |

Table B.6: GRTr predictions with the most distant generated/retrieved candidates — MetaL-WOz cross-task (in bold is the model's final response)

| | | |
|---|---|---|
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I am looking for a place to dine. The restaurant should be expensive and should serve Chinese food. |
| | Wiz | Would you like the restaurant in a specific place? |
| **Gold response** | Usr | Not in any specific area please. I need to book a table for 3 at 14:00 on Saturday. |
| **Generated candidate** | Usr | Do you have the address? [−**1.633**] |
| **Retrieved candidate** | Usr | I am looking for a restaurant in the centre of town, please. [−1.635] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I need a hotel in the south of the city, it is ok if the hotel doesn't has free parking. |
| | Wiz | I have one hotel that meets your needs . It 's the Lensfield Hotel that offers free wifi and parking . They have a 3 star rating . Would you like more info on it ? |
| | Usr | What is the price of the Lensfield? |
| | Wiz | The Lensfield Hotel is expensive. |
| | Usr | Is there a moderate place in the in the south that has free parking? |
| | Wiz | No, no hotels meet your criteria. |
| | Usr | Are there any moderately priced hotels that don't have free parking, but have wifi? |
| | Wiz | We have 2 locations that are in the south , moderately priced , and have free wifi . They are both guesthouses , would one of those meet your needs ? |
| **Gold response** | Usr | Yes, can you give me the address of your favourite one? |
| **Generated candidate** | Usr | Can you book me for 2 nights for 2 nights starting from Wednesday. [−0.013] |
| **Retrieved candidate** | Usr | Thanks for the information! [−**0.01**] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I am hoping you might be able to help me find a place to stay last minute . I am on a bit of a budget so I need something moderate . |
| | Wiz | I have three hotels available. The Ashley Hotel, Cityroomz, and Lovell Lodge. Do you need parking? |
| | Usr | I need one in the centre and has free wifi. |
| | Wiz | Then Cityroomz is just for you! It is located at sleeperz hotel, station road. Do you need more information or do you want to book a room ? |
| **Gold response** | Usr | That is all that I need. Thank you so much. |
| **Generated candidate** | Usr | I would like to book it for 4 people and 1 night starting on Saturday. [−**0.624**] |
| **Retrieved candidate** | Usr | Okay, I 'd like to book a room at the Gonville Hotel for 4 nights. There will be 6 people and we will be arriving on Saturday. [−**0.628**] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I am looking for information about the Parkside Police Station. |
| | Wiz | Sure, Parkside Police Station Telephone 01223358966, Address Parkside, Cambridge, Postcode CB11JG. Is there anything else I can do for you today ? |
| **Gold response** | Usr | No, that's everything I needed, thanks. |
| **Generated candidate** | Usr | No, that is all, thank you. [**2.783**] |
| **Retrieved candidate** | Usr | Yes, I 'll also need their phone number. [2.779] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I'm looking for a place to dine that serves British food. It should be in the moderate price range. |
| | Wiz | How about The Oak Bistro? It's located in the Centre. Would you like more information or to book a table? |
| **Gold response** | Usr | The Oak Bistro sounds fabulous. Can I get a table there on Sunday at 18:30? There will be 6 of us. |
| **Generated candidate** | Usr | Yes, I need to book a table for 5 people at 15:30 on saturday. [**1.259**] |
| **Retrieved candidate** | Usr | Yes, I 'd like a table for a party of 8 at 11:15 on Thursday. [1.255] |

TABLE B.7: GRTR predictions with the closest generated/retrieved candidates — MULTIWOZ (in bold is the model's final response)

| | | |
|---|---|---|
| **Context** | Wiz | Hello how may I help you? |
| | Usr | i am looking for a train. The train should leave on Monday and should arrive by 12:45. |
| | Wiz | Where would you like to go? |
| **Gold response** | Usr | I would like to depart from Cambridge, and travel to Bishops Stortford. |
| **Generated candidate** | Usr | i would like to go to Cambridge. [**5**.**253**] |
| **Retrieved candidate** | Usr | I would like to leave after 09:00 on Sunday. [−11.019] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I'm looking for a restaurant in the west side that's in the moderate price range. |
| | Wiz | I have three listings, one for british, indian, italian. Are any of those types of food you are looking for? |
| | Usr | Are any of them available for 8 people at 19:00 on Wednesday? |
| | Wiz | There are two restaurants available the Indian restaurant Meghna and the Italian restaurant Prezzo 8 people at 19:00 on Wednesday . Which one do you prefer ? |
| **Gold response** | Usr | The Italian restaurant Prezzo sounds good. |
| **Generated candidate** | Usr | Prezzo 8 will be great, how about Meghna? [**7**.**966**] |
| **Retrieved candidate** | Usr | Let's try the Brasserie. [−6.55] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | Hello there. I am trying to locate a train for Monday. |
| | Wiz | Sure. Where will you be going? |
| | Usr | I am going to Cambridge. |
| | Wiz | All right, and where would you like to depart from? |
| | Usr | I 'll will be leaving from London Liverpool Street. |
| | Wiz | I've got 10 trains that are on that route. Do you have a preferred departure or arrival time? |
| | Usr | I need on that will leave London after 21:00. |
| | Wiz | I have two trains that will leave after 21:00 . One leaves at 21:39 and one leaves at 23:39 . Which would you prefer ? |
| **Gold response** | Usr | What time does the one that leaves at 21:39 arrive in Cambridge? Also, what is the train ID? |
| **Generated candidate** | Usr | I 'd like to leave after 21:00 and leave the train that leaves after 21:00 [−12.611] |
| **Retrieved candidate** | Usr | I don't actually need to book at the moment. I just need to know the departure time and travel time for this train, if you don't [**0**.**699**] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | I am looking for city centre north b and b |
| | Wiz | I have found the guesthouse you were wanting. Would you like me to book this for you? |
| **Gold response** | Usr | Yes, please book it for 1 person and for 5 nights starting Friday. |
| **Generated candidate** | Usr | Yes please book it for 5 people. [**0**.**706**] |
| **Retrieved candidate** | Usr | Do any of those include free parking? [−11.545] |
| **Context** | Wiz | Hello how may I help you? |
| | Usr | Hi, I'm looking for some train information. Could you tell me what trains leave on Wednesday for Norwich? |
| | Wiz | There are 19 entries found. Where would you be coming from? |
| | Usr | I 'll be departing from Cambridge and I need to arrive by 12:00. |
| | Wiz | There is a train that arrives at 11:55. The trainID is TR9635. Would you like me to book that? |
| **Gold response** | Usr | Sure, that sounds great. |
| **Generated candidate** | Usr | Yes, that would be great. [**4**.**836**] |
| **Retrieved candidate** | Usr | Great can I get TR5173 booked for 3 people please? [−7.307] |

TABLE B.8: GRTR predictions with the most distant generated/retrieved candidates — MULTIWOZ (in bold is the model's final response)

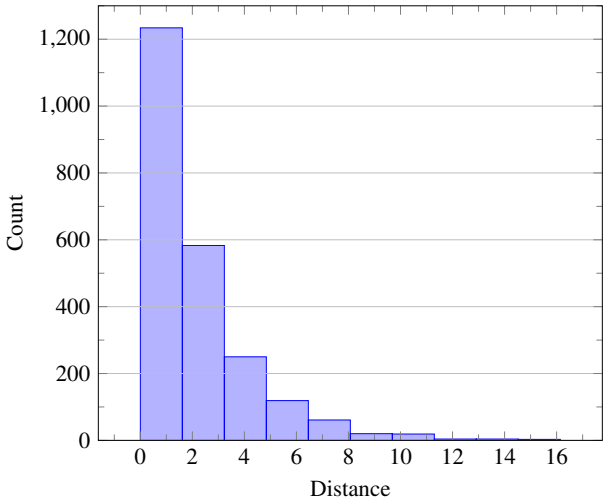



FIGURE B.1: METALWOZ pure task: a histogram of pairwise distances between generated and retrieved GRTR candidates

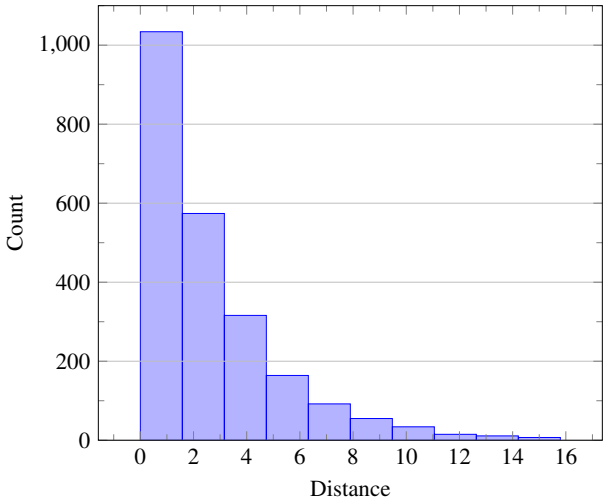



FIGURE B.2: METALWOZ cross-task: a histogram of pairwise distances between generated and retrieved GRTR candidates

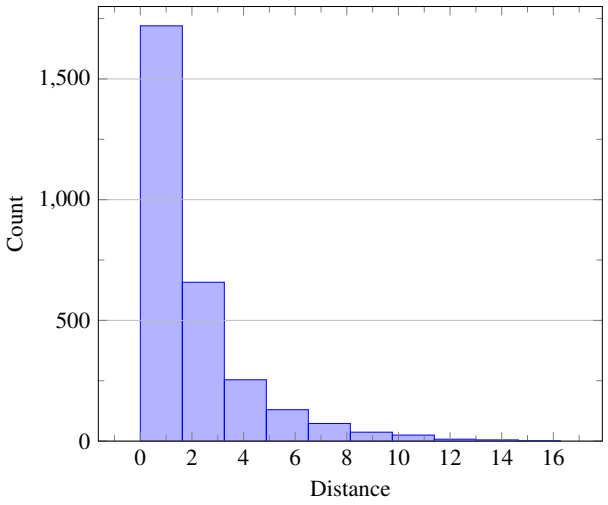



FIGURE B.3: MULTIWOZ: a histogram of pairwise distances between generated and retrieved GRTR candidates

# Appendix C

# Disfluency Detection — Supplementary Material

| Parameter | Value |
|---|---|
| Optimiser | stochastic gradient descent |
| Loss function | weighted cross-entropy |
| Vocabulary size | 6157 |
| Embedding size | 128 |
| MLP layer sizes | [128] |
| Learning rate | 0.01 |
| Learning rate decay | 0.9 |
| Batch size | 32 |
| $\alpha$ | 0.1 |
| $\lambda$ | 0.001 |
| $\gamma$ | 1.05 |

TABLE C.1: Multi-task LSTM training setup

| Label type | Label | Frequency |
|---|---|---|
| Fluent token | <f/> | 574 771 |
| Edit token | <e/> | 45 729 |
| Single-token substitution | <rm-*{1-8}*/><rpEndSub/> | 13 003 |
| Single-token deletion | <rm-*{1-8}*/><rpEndDel/> | 1011 |
| Multi-token substitution start | <rm-*{1-8}*/><rpMid/> | 6976 |
| Multi-token substitution end | <rpEndSub> | 6818 |

TABLE C.2: SWDA labels

# Appendix D

# Data-Efficiency in Social Dialogue — Supplementary Material

| Parameter | Value |
|---|---|
| vocabulary size | 60000 |
| learning rate | 0.01 |
| embedding size | 256 |
| RNN cell type | GRU |
| optimiser | Adagrad |
| loss function | MSE |
| dropout ratio | 0.4 |
| predictor layer sizes | [256] (length), [128, 32, 32] (rating) |
| batch size | 8 |
| max utterance length | 30 tokens |

TABLE D.1: Neural rankers training setup

| **Parameter** | **Value** |
|---|---|
| feature bit length | 16 |
| loss function | squared |
| features | context ngrams, response ngrams,<br>turn number, bot, utterance length,<br>handcrafted features |
| quadratic features | response ngrams × response ngrams,<br>context ngrams × response ngrams,<br>bot name × response ngrams,<br>bot name × context ngrams,<br>bot name × handcrafted features |
| cubic features | bot name × context ngrams × response ngrams |
| holdout set | off |
| passes number | 1 |

TABLE D.2: VowpalWabbit ranker training setup

# Bibliography


Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Agarwal, A., Chapelle, O., Dudík, M., and Langford, J. (2014). A reliable effective terascale linear learning system. *Journal of Machine Learning Research*, 15(1):1111–1133.

Aist, G., Allen, J., Campana, E., Gallo, C. G., Stoness, S., Swift, M., and Tanenhaus, M. K. (2007). Incremental dialogue system faster than and preferred to its nonincremental counterpart. In *Annual Conference of the Congitive Science Society*.

Al-Rfou, R., Pickett, M., Snaider, J., Sung, Y.-h., Strope, B., and Kurzweil, R. (2016). Conversational Contextual Cues: The Case of Personalization and History for Response Ranking. *CoRR*, abs/1606.00372.

Allen, J. F., Byron, D. K., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. (2001). Toward conversational human-computer interaction. *AI Mag.*, 22(4):27–37.

Asri, L. E., Schulz, H., Sharma, S., Zumer, J., Harris, J., Fine, E., Mehrotra, R., and Suleman, K. (2017). Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, pages 207–219.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., and Blackburn, J. (2020). The pushshift reddit dataset. *CoRR*, abs/2001.08435.

Bellman, R. and Kalaba, R. (1957). On the role of dynamic programming in statistical communication theory. *IRE Trans. Information Theory*, 3(3):197–203.

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., de Oliveira Pinto, H. P., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., and Zhang, S. (2019). Dota 2 with large scale deep reinforcement learning. *CoRR*, abs/1912.06680.

Blanco, R., Ottaviano, G., and Meij, E. (2015). Fast and space-efficient entity linking in queries. In *Proceedings of the Eight ACM International Conference on Web Search and Data Mining*, WSDM 15, New York, NY, USA. ACM.

Bordes, A., Boureau, Y., and Weston, J. (2017). Learning end-to-end goal-oriented dialog. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Bortfeld, H., Leon, S. D., Bloom, J. E., Schober, M. F., and Brennan, S. E. (2001). Disfluency rates in conversation: Effects of age, relationship, topic, role, and gender. *Language and speech*, 44(2):123–147.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186, Heidelberg. Physica-Verlag HD.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Józefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 10–21.

Brennan, S. and Schober, M. (2001). How listeners compensate for disfluencies in spontaneous speech. *Journal of Memory and Language*, 44(2):274–296.

Budzianowski, P. and Vulic, I. (2019). Hello, it's GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP 2019, Hong Kong, November 4, 2019*, pages 15–22. Association for Computational Linguistics.

Budzianowski, P., Wen, T., Tseng, B., Casanueva, I., Ultes, S., Ramadan, O., and Gasic, M. (2018). Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural*

*Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026. Association for Computational Linguistics.

Burges, C. J. C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. N. (2005). Learning to rank using gradient descent. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, pages 89–96.

Cann, R., Kempson, R., and Marten, L. (2005). *The Dynamics of Language*. Elsevier, Oxford.

Cao, K. and Clark, S. (2017). Latent variable dialogue models and their diversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 182–187. Association for Computational Linguistics.

Chang, A. X. and Manning, C. D. (2012). Sutime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 3735–3740.

Charniak, E. and Johnson, M. (2001). Edit detection and parsing for transcribed speech. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–9. Association for Computational Linguistics.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014a). On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014b). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734.

Colman, M. and Healey, P. G. T. (2011). The distribution of repair in dialogue. In *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society*, pages 1563–1568, Boston, MA.

Cooper, R. (2005). Records and record types in semantic theory. *Journal of Logic and Computation*, 15(2):99–112.

Cuayáhuitl, H. (2016). Simpleds: A simple deep reinforcement learning dialogue system. In *Dialogues with Social Robots - Enablements, Analyses, and Evaluation, Seventh International Workshop on Spoken Dialogue Systems, IWSDS 2016, Saariselkä, Finland, January 13-16, 2016*, volume 427 of *Lecture Notes in Electrical Engineering*, pages 109–118. Springer.

Curry, A. C., Papaioannou, I., Suglia, A., Agarwal, S., Shalyminov, I., Xu, X., Dusek, O., Eshghi, A., Konstas, I., Rieser, V., and Lemon, O. (2018). Alana v2: Entertaining and informative open-domain social dialogue using ontologies and entity linking. In *2018 Alexa Prize Proceedings*.

Dahlbäck, N., Jönsson, A., and Ahrenberg, L. (1998). *Wizard of Oz Studies—Why and How*, page 610–619. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Danescu-Niculescu-Mizil, C. and Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society.

Dethlefs, N. and Cuayáhuitl, H. (2010). Hierarchical reinforcement learning for adaptive text generation. In *INLG 2010 - Proceedings of the Sixth International Natural Language Generation Conference, July 7-9, 2010, Trim, Co. Meath, Ireland*. The Association for Computer Linguistics.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Dinarelli, M., Quarteroni, S., Tonelli, S., Moschitti, A., and Riccardi, G. (2009). Annotating spoken dialogs: From speech segments to dialog acts and frame semantics. In *Proceedings of SRSL 2009, the 2nd Workshop on Semantic Representation of Spoken Language*, pages 34–41, Athens, Greece. Association for Computational Linguistics.

Duchi, J. C., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Dusek, O. and Jurcicek, F. (2015). Training a natural language generator from unaligned data. *Proceedings of ACL*.

Dusek, O. and Jurcícek, F. (2016). Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

Dziri, N., Kamalloo, E., Mathewson, K. W., and Zaïane, O. R. (2019). Evaluating coherence in dialogue systems using entailment. In *NAACLHLT*.

E, H., Niu, P., Chen, Z., and Song, M. (2019). A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5467–5471. Association for Computational Linguistics.

Elman, J. L. (1990). Finding structure in time. *Cogn. Sci.*, 14(2):179–211.

Eric, M., Goel, R., Paul, S., Sethi, A., Agarwal, S., Gao, S., and Hakkani-Tur, D. (2019). Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Eric, M., Krishnan, L., Charette, F., and Manning, C. D. (2017). Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, pages 37–49.

Eric, M. and Manning, C. D. (2017). A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. In Lapata, M., Blunsom, P., and Koller, A., editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 468–473. Association for Computational Linguistics.

Eshghi, A. (2015). DS-TTR: An incremental, semantic, contextual parser for dialogue. In *Proceedings of Semdial 2015 (goDial), the 19th workshop on the semantics and pragmatics of dialogue*.

Eshghi, A., Hough, J., and Purver, M. (2013a). Incremental grammar induction from child-directed dialogue utterances. In *Proceedings of the 4th Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, pages 94–103, Sofia, Bulgaria. Association for Computational Linguistics.

Eshghi, A., Hough, J., Purver, M., Kempson, R., and Gregoromichelaki, E. (2012). Conversational interactions: Capturing dialogue dynamics. In *From Quantification to Conversation: Festschrift for Robin Cooper on the occasion of his 65th birthday*, volume 19 of *Tributes*, pages 325–349. College Publications, London.

Eshghi, A., Howes, C., Gregoromichelaki, E., Hough, J., and Purver, M. (2015). Feedback in conversation as incremental semantic update. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS 2015)*, London, UK. Association for Computational Linguisitics.

Eshghi, A. and Lemon, O. (2014). How domain-general can we be? Learning incremental dialogue systems without dialogue acts. In *Proceedings of Semdial 2014 (DialWatt)*.

Eshghi, A., Purver, M., and Hough, J. (2011). Dylan: Parser for dynamic syntax. Technical report, Queen Mary University of London.

Eshghi, A., Purver, M., Hough, J., and Sato, Y. (2013b). Probabilistic grammar induction in an incremental semantic framework. In *CSLP, Lecture Notes in Computer Science*. Springer.

Eshghi, A., Shalyminov, I., and Lemon, O. (2017). Bootstrapping incremental dialogue systems from minimal data: the generalisation power of dialogue grammars. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2220–2230.

Fedorenko, D., Smetanin, N., and Rodichev, A. (2018). Avoiding Echo-Responses in a Retrieval-Based Conversation System. In *Artificial Intelligence and Natural Language*.

Fernández, R. (2006). *Non-Sentential Utterances in Dialogue: Classification, Resolution and Use*. PhD thesis, King's College London, University of London.

Fernández, R. and Ginzburg, J. (2002). Non-sentential utterances: Grammar and dialogue dynamics in corpus annotation. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, pages 253–259.

Ferreira, F., Lau, E. F., and Bailey, K. G. D. (2004). Disfluencies, language comprehension, and tree adjoining grammars. *Cognitive Science*, 28(5):721–749.

Finkel, J. R., Grenager, T., and Manning, C. D. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 363–370. The Association for Computer Linguistics.

Ganhotra, J. and Polymenakos, L. (2018). Knowledge-based end-to-end memory networks. *CoRR*, abs/1804.08204.

Gao, J., Galley, M., and Li, L. (2019). Neural approaches to conversational ai. *Foundations and Trends in Information Retrieval*.

Gilbert, C. J. and Hutto, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International AAAI Conference on Weblogs and Social Media*, pages 216–225, Ann Arbor, MI, USA.

Ginzburg, J. (2012). *The Interactive Stance: Meaning for Conversation*. Oxford University Press.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Gu, J., Wang, Y., Chen, Y., Li, V. O. K., and Cho, K. (2018). Meta-learning for low-resource neural machine translation. In *EMNLP*.

Hall, P., Miller, H., et al. (2009). Using the bootstrap to quantify the authority of an empirical ranking. *The Annals of Statistics*, 37(6B):3929–3959.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Healey, P. G. T., Mills, G., Eshghi, A., and Howes, C. (2018). Running repairs: Coordinating meaning in dialogue. *topiCS*, 10(2):367–388.

Henderson, M., Thomson, B., and Williams, J. D. (2014a). The second dialog state tracking challenge. In *Proceedings of the SIGDIAL 2014 Conference, The 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 18-20 June 2014, Philadelphia, PA, USA*, pages 263–272.

Henderson, M., Thomson, B., and Williams, J. D. (2014b). The third dialog state tracking challenge. In *2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, December 7-10, 2014*, pages 324–329. IEEE.

Henderson, M., Thomson, B., and Young, S. (2014c). Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Henderson, M., Thomson, B., and Young, S. J. (2014d). Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, December 7-10, 2014*, pages 360–365. IEEE.

Henderson, M., Thomson, B., and Young, S. J. (2014e). Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the SIGDIAL 2014 Conference, The 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 18-20 June 2014, Philadelphia, PA, USA*, pages 292–299.

Hill, F., Cho, K., and Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1367–1377.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hoffman, M., Bach, F. R., and Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems 23*, pages 856–864.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020). The curious case of neural text degeneration. *International Conference on Learning Representations*.

Hough, J. (2011). Incremental semantics driven natural language generation with self-repairing capability. In *Recent Advances in Natural Language Processing (RANLP)*, pages 79–84, Hissar, Bulgaria.

Hough, J. (2014). *Modelling incremental self-repair processing in dialogue*. PhD thesis, Queen Mary University of London, UK.

Hough, J. and Purver, M. (2014a). Probabilistic type theory for incremental dialogue processing. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 80–88, Gothenburg, Sweden. Association for Computational Linguistics.

Hough, J. and Purver, M. (2014b). Strongly incremental repair detection. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 78–89.

Hough, J. and Schlangen, D. (2015). Recurrent neural networks for incremental disfluency detection. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 849–853.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339. Association for Computational Linguistics.

Howes, C., Healey, P. G. T., and Purver, M. (2010). Tracking lexical and syntactic alignment in conversation. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, Portland, OR.

Huang, P., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. P. (2013). Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2333–2338. ACM.

Ippolito, D., Kriz, R., Sedoc, J., Kustikova, M., and Callison-Burch, C. (2019). Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, pages 3752–3762. Association for Computational Linguistics.

Ji, Z., Lu, Z., and Li, H. (2014). An Information Retrieval Approach to Short Text Conversation. *CoRR*, abs/1408.6988.

Jia, R. and Liang, P. (2016). Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Jiang, S. and de Rijke, M. (2018). Why are sequence-to-sequence models so dull? understanding the low-diversity problem of chatbots. In *Proceedings of the 2nd International Workshop on Search-Oriented Conversational AI, SCAI@EMNLP 2018, Brussels, Belgium, October 31, 2018*, pages 81–86. Association for Computational Linguistics.

Johnson, J., Douze, M., and Jégou, H. (2017). Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77.

Jurafsky, D. and Martin, J. H. (2019). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 3rd edition draft*.

Kempson, R., Gregoromichelaki, E., Eshghi, A., and Hough, J. (2017). Ellipsis in Dynamic Syntax. In *Oxford Handbook of Ellipsis*. Oxford University Press.

Kempson, R., Meyer-Viol, W., and Gabbay, D. (2001). *Dynamic Syntax: The Flow of Language Understanding*. Wiley-Blackwell.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In Hearst, M. A. and Ostendorf, M., editors, *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics.

Krause, B., Damonte, M., Dobre, M., Duma, D., Fainberg, J., Fancellu, F., Kahembwe, E., Cheng, J., and Webber, B. (2017). Edina: Building an Open Domain Socialbot with Self-dialogues. In *1st Proceedings of Alexa Prize*, Las Vegas, NV, USA. arXiv: 1709.09816.

Kumar, G., Henderson, M., Chan, S., Nguyen, H., and Ngoo, L. (2018). Question-answer selection in user to user marketplace conversations. In *9th International Workshop on Spoken Dialogue System Technology, IWSDS 2018, Singapore, April 18-20, 2018*, volume 579 of *Lecture Notes in Electrical Engineering*, pages 397–403. Springer.

Kwiatkowski, T., Choi, E., Artzi, Y., and Zettlemoyer, L. (2013). Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lane, I., Kawahara, T., Matsui, T., and Nakamura, S. (2007). Out-of-domain utterance detection using classification confidences of multiple topics. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):150–161.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324.

Lee, S. (2017). Toward continual learning for conversational agents. *CoRR*, abs/1712.09943.

Lee, S., Schulz, H., Atkinson, A., Gao, J., Suleman, K., El Asri, L., Adada, M., Huang, M., Sharma, S., Tay, W., and Li, X. (2019a). Multi-domain task-completion dialog challenge. In *Dialog System Technology Challenges 8*.

Lee, S. and Shalyminov, I. (2019). Contextual out-of-domain utterance handling with counterfeit data augmentation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7205–7209.

Lee, S., Zhu, Q., Takanobu, R., Li, X., Zhang, Y., Zhang, Z., Li, J., Peng, B., Li, X., Huang, M., and Gao, J. (2019b). Convlab: Multi-domain end-to-end dialog system platform. In *ACL*.

Lewis, M. and Steedman, M. (2013). Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2016a). A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics.

Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J. (2016b). Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1192–1202.

Li, J., Peng, B., Lee, S., Gao, J., Takanobu, R., Zhu, Q., Huang, M., Schulz, H., Atkinson, A., and Adada, M. (2020). Results of the multi-domain task-completion dialog challenge. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, Eighth Dialog System Technology Challenge Workshop*.

Liang, W. and Yang, M. (2018). Hierarchical hybrid code networks for task-oriented dialogue. In *Intelligent Computing Theories and Application*, pages 194–204. Springer International Publishing.

Lison, P. and Tiedemann, J. (2016). Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Liu, H., Lin, T., Sun, H., Lin, W., Chang, C.-W., Zhong, T., and Rudnicky, A. (2017). RubyStar: A Non-Task-Oriented Mixture Model Dialog System. In *1st Proceedings of Alexa Prize*, Las Vegas, NV, USA. arXiv: 1711.02781.

Liu, T.-Y. (2011). *Learning to Rank for Information Retrieval*. Springer.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Lowe, R., Noseworthy, M., Serban, I. V., Angelard-Gontier, N., Bengio, Y., and Pineau, J. (2017). Towards an automatic turing test: Learning to evaluate dialogue responses. In *ACL*.

Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015). The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of SIGDIAL*, pages 285–294.

Lu, Y., Keung, P., Zhang, S., Sun, J., and Bhardwaj, V. (2017). A practical approach to dialogue response generation in closed domains. *CoRR*, abs/1703.09439.

Luo, L., Huang, W., Zeng, Q., Nie, Z., and Sun, X. (2019). Learning personalized end-to-end goal-oriented dialog. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6794–6801. AAAI Press.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421. The Association for Computational Linguistics.

Ma, Y., Zeng, Z., Zhu, D., Li, X., Yang, Y., Yao, X., Zhou, K., and Shen, J. (2019). An end-to-end dialogue state tracking system with machine reading comprehension and wide & deep classification. *CoRR*, abs/1912.09297.

Mairesse, F., Jurčíček, F., Keizer, S., Thomson, B., Yu, K., and Young, S. (2009). Spoken language understanding from unaligned data using discriminative classification models. In *Proceedings of ICASSP*.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Maystre, L. and Grossglauser, M. (2017). Just sort it! A simple and effective approach to active preference learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2344–2353. PMLR.

Merity, S., Keskar, N. S., and Socher, R. (2018). Regularizing and optimizing LSTM language models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2017). Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Merriam-Webster Dictionary (2020). Dialogue — Merriam-Webster.com dictionary. [Online; accessed 15-May-2020].

Meteer, M., Taylor, A., MacIntyre, R., and Iyer, R. (1995). Disfluency annotation stylebook for the switchboard corpus.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In Kobayashi, T., Hirose, K., and Nakamura, S., editors, *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048. ISCA.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Miller, A. H., Feng, W., Batra, D., Bordes, A., Fisch, A., Lu, J., Parikh, D., and Weston, J. (2017). Parlai: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017 - System Demonstrations*, pages 79–84. Association for Computational Linguistics.

Miller, A. H., Fisch, A., Dodge, J., Karimi, A., Bordes, A., and Weston, J. (2016). Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1400–1409. The Association for Computational Linguistics.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A. (2013). Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602.

Mrksic, N., Séaghdha, D. Ó., Thomson, B., Gasic, M., Su, P., Vandyke, D., Wen, T., and Young, S. J. (2015). Multi-domain dialog state tracking using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 794–799. The Association for Computer Linguistics.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814. Omnipress.

Nakano, M., Sato, S., Komatani, K., Matsuyama, K., Funakoshi, K., and Okuno, H. G. (2011). A two-stage domain selection framework for extensible multi-domain spoken dialogue systems. In *Proceedings of the SIGDIAL 2011 Conference*, pages 18–29. Association for Computational Linguistics.

Nalisnick, E. T., Matsukawa, A., Teh, Y. W., Görür, D., and Lakshminarayanan, B. (2019). Do deep generative models know what they don't know? In *7th International Conference*

*on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Novikova, J., Dusek, O., Cercas-Curry, A., and Rieser, V. (2017). Why we need new evaluation metrics for nlg. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Olah, C. and Carter, S. (2016). Attention and augmented recurrent neural networks. *Distill*.

Oviatt, S. (1995). Predicting spoken disfluencies during human–computer interaction. *Computer Speech & Language*, 9(1):19–35.

Papaioannou, I., Curry, A. C., Part, J. L., Shalyminov, I., Xu, X., Yu, Y., Dusek, O., Rieser, V., and Lemon, O. (2017). An ensemble model with ranking for social dialogue. In *NIPS Workshop on Conversational AI*.

Pappu, A., Blanco, R., Mehdad, Y., Stent, A., and Thadani, K. (2017). Lightweight multilingual entity extraction and linking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM 17, New York, NY, USA. ACM.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS-W*.

Peckham, J. (1991). Speech understanding and dialogue over the telephone: an overview of the ESPRIT SUNDIAL project. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543.

Peters, M. E., Ammar, W., Bhagavatula, C., and Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1756–1765.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Pickering, M. J., Clifton, C., and Crocker, M. W. (1999). *Architectures and Mechanisms in Sentence Comprehension*, page 1–28. Cambridge University Press.

Purver, M., Eshghi, A., and Hough, J. (2011). Incremental semantic construction in a dialogue system. In *Proceedings of the 9th International Conference on Computational Semantics*, pages 365–369, Oxford, UK.

Purver, M., Gregoromichelaki, E., Meyer-Viol, W., and Cann, R. (2010). Splitting the 'I's and crossing the 'You's: Context, speech acts and grammar. In *Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue*, pages 43–50, Poznań. Polish Society for Cognitive Science.

Purver, M., Hough, J., and Gregoromichelaki, E. (2014). Dialogue and compound contributions. In *Natural Language Generation in Interactive Systems*, pages 63–92. Cambridge University Press.

Purver, M., Hough, J., and Howes, C. (2018). Computational models of miscommunication phenomena. In *Topics in Cognitive Science (topiCS)*, volume 10.

Qian, K. and Yu, Z. (2019). Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2639–2649.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2018). Language models are unsupervised multitask learners.

Ram, A., Prasad, R., Khatri, C., Venkatesh, A., Gabriel, R., Liu, Q., Nunn, J., Hedayatnia, B., Cheng, M., Nagar, A., King, E., Bland, K., Wartick, A., Pan, Y., Song, H., Jayadevan, S., Hwang, G., and Pettigrue, A. (2017). Conversational AI: The Science Behind the Alexa Prize. In *1st Proceedings of Alexa Prize*, Las Vegas, NV, USA. arXiv: 1801.03604.

Ramachandran, D. and Ratnaparkhi, A. (2015). Belief tracking with stacked relational trees. In *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, pages 68–76. The Association for Computer Linguistics.

Rei, M. (2017). Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 2121–2130.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org.

Rieser, V. and Lemon, O. (2009). Natural language generation as planning under uncertainty for spoken dialogue systems. *Proceedings of ACL*.

Rieser, V. and Lemon, O. (2011). *Reinforcement Learning for Adaptive Dialogue Systems - A Data-driven Methodology for Dialogue Management and Natural Language Generation*. Theory and Applications of Natural Language Processing. Springer.

Rojas-Barahona, L. M., Gasic, M., Mrksic, N., Su, P., Ultes, S., Wen, T., Young, S. J., and Vandyke, D. (2017). A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 438–449.

Rudnicky, A. I., Thayer, E. H., Constantinides, P. C., Tchou, C., Shern, R., Lenzo, K. A., Xu, W., and Oh, A. (1999). Creating natural dialogs in the carnegie mellon communicator system. In *Sixth European Conference on Speech Communication and Technology, EUROSPEECH 1999, Budapest, Hungary, September 5-9, 1999*. ISCA.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). *Learning Representations by Back-Propagating Errors*, page 696–699. MIT Press, Cambridge, MA, USA.

Ryu, S., Kim, S., Choi, J., Yu, H., and Lee, G. G. (2017). Neural sentence embedding using only in-domain sentences for out-of-domain sentence detection in dialog systems. *Pattern Recognition Letters*, 88:26–32.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Schapire, R. and Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. *Machine Learning - ML*, 39:135–168.

Schlangen, D. and Hough, J. (2017). Joint, incremental disfluency detection and utterance segmentation from speech. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 326–336.

Schlangen, D. and Skantze, G. (2009). A general, abstract model of incremental dialogue processing. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 710–718, Athens, Greece. Association for Computational Linguistics.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Serban, I. V., Lowe, R., Henderson, P., Charlin, L., and Pineau, J. (2018). A survey of available corpora for building data-driven dialogue systems: The journal version. *Dialogue Discourse*, 9(1):1–49.

Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Ke, N. R., Mudumba, S., de Brébisson, A., Sotelo, J., Suhubdy, D., Michalski, V., Nguyen, A., Pineau, J., and Bengio, Y. (2017a). A deep reinforcement learning chatbot. *NIPS Workshop on Conversational AI*, abs/1709.02349.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 3776–3783. AAAI Press.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A. C., and Pineau, J. (2015). Hierarchical neural network generative models for movie dialogues. *CoRR*, abs/1507.04808.

Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A. C., and Bengio, Y. (2017b). A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3295–3301. AAAI Press.

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, page 134–141, USA. Association for Computational Linguistics.

Shah, P., Hakkani-Tür, D., Tür, G., Rastogi, A., Bapna, A., Nayak, N., and Heck, L. (2018). Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Shalyminov, I., Dusek, O., and Lemon, O. (2018a). Neural response ranking for social conversation: A data-efficient approach. In *Proceedings of the 2nd International Workshop on Search-Oriented Conversational AI, SCAI@EMNLP 2018, Brussels, Belgium, October 31, 2018*, pages 1–8.

Shalyminov, I., Eshghi, A., and Lemon, O. (2017). Challenging Neural Dialogue Models with Natural Data: Memory Networks Fail on Incremental Phenomena. In *Proceedings of the 21st Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2017 - SaarDial)*.

Shalyminov, I., Eshghi, A., and Lemon, O. (2018b). Multi-Task Learning for Domain-General Spoken Disfluency Detection in Dialogue Systems. In *Proceedings of the 22st Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2018 - AixDial)*.

Shalyminov, I. and Lee, S. (2018). Improving robustness of neural dialog systems in a data-efficient way with turn dropout. *The Second NeurIPS Workshop on Conversational AI: "Today's Practice and Tomorrow's Potential"*, abs/1811.12148.

Shalyminov, I., Lee, S., Eshghi, A., and Lemon, O. (2019a). Data-efficient goal-oriented conversation with dialogue knowledge transfer networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1741–1751. Association for Computational Linguistics.

Shalyminov, I., Lee, S., Eshghi, A., and Lemon, O. (2019b). Few-shot dialogue generation without annotated data: A transfer learning approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 32–39, Stockholm, Sweden. Association for Computational Linguistics.

Shalyminov, I., Sordoni, A., Atkinson, A., and Schulz, H. (2020). Fast domain adaptation for goal-oriented dialogue using a hybrid generative-retrieval transformer. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8039–8043.

Shalyminov, I., Sordoni, A., Atkinson, A., and Schulz, H. (2020). Hybrid generative-retrieval transformers for dialogue domain adaptation. *Dialog State Tracking Challenge 8, DSTC8AAAI 2020*.

Sharma, S., El Asri, L., Schulz, H., and Zumer, J. (2017). Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.

Shi, H., Ushio, T., Endo, M., Yamagami, K., and Horii, N. (2016). A multichannel convolutional neural network for cross-language dialog state tracking. In *2016 IEEE Spoken Language Technology Workshop, SLT 2016, San Diego, CA, USA, December 13-16, 2016*, pages 559–564.

Shi, W., Qian, K., Wang, X., and Yu, Z. (2019a). How to build user simulators to train rl-based dialog systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1990–2000. Association for Computational Linguistics.

Shi, W., Zhao, T., and Yu, Z. (2019b). Unsupervised dialog structure learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1797–1807. Association for Computational Linguistics.

Shriberg, E. (1994). *Preliminaries to a Theory of Speech Disfluencies*. PhD thesis, University of California, Berkeley.

Shukla, S., Liden, L., Shayandeh, S., Kamal, E., Li, J., Mazzola, M., Park, T., Peng, B., and Gao, J. (2020). Conversation learner - A machine teaching tool for building dialog managers for task-oriented dialog systems. In Çelikyilmaz, A. and Wen, T., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 343–349. Association for Computational Linguistics.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Singh, S. P., Litman, D. J., Kearns, M. J., and Walker, M. A. (2002). Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *J. Artif. Intell. Res.*, 16:105–133.

Skantze, G. and Hjalmarsson, A. (2010). Towards incremental speech generation in dialogue systems. In *Proceedings of the SIGDIAL 2010 Conference*, pages 1–8, Tokyo, Japan. Association for Computational Linguistics.

Song, Y., Yan, R., Li, X., Zhao, D., and Zhang, M. (2016). Two are better than one: An ensemble of retrieval- and generation-based dialog systems. *CoRR*, abs/1610.07149.

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., and Dolan, B. (2015). A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado. Association for Computational Linguistics.

Su, P., Budzianowski, P., Ultes, S., Gasic, M., and Young, S. J. (2017). Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, pages 147–157. Association for Computational Linguistics.

Su, P., Gasic, M., Mrksic, N., Rojas-Barahona, L. M., Ultes, S., Vandyke, D., Wen, T., and Young, S. J. (2016). Continuously learning neural dialogue management. *CoRR*, abs/1606.02689.

Sukhbaatar, S., Szlam, A., Weston, J., and Fergus, R. (2015). End-to-end memory networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2440–2448.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference*

*on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Tan, M., Xiang, B., and Zhou, B. (2015). Lstm-based deep learning models for non-factoid answer selection. *CoRR*, abs/1511.04108.

Tiedemann, J. (2009). News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Tur, G. (2011). *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley and Sons.

Tur, G., Deoras, A., and Hakkani-Tür, D. (2014). Detecting out-of-domain utterances addressed to a virtual personal assistant. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Turing, A. M. (1950). I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Venkatesh, A., Khatri, C., Ram, A., Guo, F., Gabriel, R., Nagar, A., Prasad, R., Cheng, M., Hedayatnia, B., Metallinou, A., Goel, R., Yang, S., and Raju, A. (2017). On Evaluating and Comparing Conversational Agents. In *NIPS 2017 Workshop on Conversational AI (ConvAI)*, Long Beach, CA, USA. arXiv: 1801.03625.

Vinyals, O., Fortunato, M., and Jaitly, N. (2015). Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700.

Vinyals, O. and Le, Q. V. (2015). A neural conversational model. *ICML*.

Wang, J., Yu, L., Zhang, W., Gong, Y., Xu, Y., Wang, B., Zhang, P., and Zhang, D. (2017a). Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 515–524, New York, NY, USA. ACM.

Wang, S., Che, W., and Liu, T. (2016). A neural attention model for disfluency detection. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 278–287.

Wang, S., Che, W., Zhang, Y., Zhang, M., and Liu, T. (2017b). Transition-based disfluency detection using lstms. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2785–2794.

Wang, Y.-Y., Deng, L., and Acero, A. (2005). Spoken language understanding. an introduction to the statistical framework. *IEEE Signal Processing Magazine*, 22(5):16–31.

Ward, W. (1991). Understanding spontaneous speech: The phoenix system. In *Proceedings of the Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference*, ICASSP '91, page 365–367, USA. IEEE Computer Society.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45.

Wen, T., Gasic, M., Kim, D., Mrksic, N., Su, P., Vandyke, D., and Young, S. J. (2015a). Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, pages 275–284. The Association for Computer Linguistics.

Wen, T.-H., Gašić, M., Mrkšić, N., Su, P.-H., Vandyke, D., and Young, S. (2015b). Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Weston, J., Chopra, S., and Bordes, A. (2015). Memory networks. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Weston, J., Dinan, E., and Miller, A. H. (2018). Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2nd International Workshop on Search-Oriented Conversational AI, SCAI@EMNLP 2018, Brussels, Belgium, October 31, 2018*, pages 87–92. Association for Computational Linguistics.

Williams, J., Kamal, E., Ashour, M., Amr, H., Miller, J., and Zweig, G. (2015). Fast and easy language understanding for dialog systems with microsoft language understanding intelligent service (luis). In *Proceedings of 2015 SIGDIAL Conference, Prague*. ACL - Association for Computational Linguistics. Demonstration at SIGDIAL 2015.

Williams, J. and Liden, L. (2017). Demonstration of interactive dialog teaching for learning a practical end-to-end dialog manager. In *Proceedings of 2017 SIGDIAL Conference*. Association for Computational Linguistics.

Williams, J. D., Asadi, K., and Zweig, G. (2017). 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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 665–677. Association for Computational Linguistics.

Williams, J. D., Raux, A., and Henderson, M. (2016). The dialog state tracking challenge series: A review. *D&D*, 7(3):4–33.

Williams, J. D., Raux, A., Ramachandran, D., and Black, A. W. (2013). The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference, The 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 22-24 August 2013, SUPELEC, Metz, France*, pages 404–413. The Association for Computer Linguistics.

Williams, J. D. and Young, S. J. (2007). Partially observable markov decision processes for spoken dialog systems. *Comput. Speech Lang.*, 21(2):393–422.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

Wolf, T., Sanh, V., Chaumond, J., and Delangue, C. (2019). Transfertransfo: A transfer learning approach for neural network based conversational agents. *NeurIPS Workshop on Conversational AI*.

Wu, B., Wang, B., and Xue, H. (2016). Ranking responses oriented to conversational relevance in chat-bots. In *Proceedings of COLING*, pages 652–662, Osaka, Japan.

Yamada, K. and Knight, K. (2001). A syntax-based statistical translation model. In *Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference, July 9-11, 2001, Toulouse, France*, pages 523–530. Morgan Kaufmann Publishers.

Yan, R., Song, Y., and Wu, H. (2016). Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System. In *Proceedings of SIGIR*, pages 55–64, Pisa, Italy.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. J., and Hovy, E. H. (2016). Hierarchical attention networks for document classification. In Knight, K., Nenkova, A., and Rambow, O., editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1480–1489. The Association for Computational Linguistics.

Young, S. (2003). Hidden vector state model for hierarchical semantic parsing. volume 1, pages I–268.

Young, S. (2009). Cued standard dialogue acts. *Internal Report, Dialogue Systems Group, Cambridge University*.

Young, S., Gasic, M., Thomson, B., and Williams, J. D. (2013). Pomdp-based statistical spoken dialogue systems: a review. *Proceedings of IEEE*, 101:1160–1179.

Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., and Yu, K. (2010). The Hidden Information State model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

Young, S. J. (2010). Still talking to machines (cognitively speaking). In Kobayashi, T., Hirose, K., and Nakamura, S., editors, *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1–10. ISCA.

Yu, Z., Papangelis, A., and Rudnicky, A. (2015). TickTock: A Non-Goal-Oriented Multimodal Dialog System with Engagement Awareness. In *Turn-Taking and Coordination in Human-Machine Interaction: Papers from the 2015 AAAI Spring Symposium*, pages 108–111, Palo Alto, CA, USA.

Yu, Z., Xu, Z., Black, A. W., and Rudnicky, A. I. (2016). Strategy and Policy Learning for Non-Task-Oriented Conversational Systems. In *Proc. SIGDIAL*, Los Angeles, CA, USA.

Zamani, H., Dehghani, M., Croft, W. B., Learned-Miller, E. G., and Kamps, J. (2018). From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 497–506. ACM.

Zang, X., Rastogi, A., Sunkara, S., Gupta, R., Zhang, J., and Chen, J. (2020). Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, ACL 2020*, pages 109–117.

Zhang, Y., Galley, M., Gao, J., Gan, Z., Li, X., Brockett, C., and Dolan, B. (2018). Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 1815–1825.

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J., and Dolan, B. (2020). Dialogpt: Large-scale generative pre-training for conversational response generation. In *ACL, system demonstration*.

Zhao, S., Song, J., and Ermon, S. (2019). Infovae: Balancing learning and inference in variational autoencoders. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5885–5892. AAAI Press.

Zhao, T. and Eskénazi, M. (2018). Zero-shot dialog generation with cross-domain latent actions. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia, July 12-14, 2018*, pages 1–10.

Zhao, T., Lee, K., and Eskénazi, M. (2018). Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1098–1107.

Zhao, T., Zhao, R., and Eskénazi, M. (2017). Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 654–664.

Zhou, L., Gao, J., Li, D., and Shum, H. (2020). The design and implementation of xiaoice, an empathetic social chatbot. *Comput. Linguistics*, 46(1):53–93.

Zhuang, Y., Wang, X., Zhang, H., Xie, J., and Zhu, X. (2018). An Ensemble Approach to Conversation Generation. In *Natural Language Processing and Chinese Computing*, volume 10619, pages 51–62. Springer International Publishing. DOI: 10.1007/978-3-319-73618-1_5.

Zilka, L. and Jurcícek, F. (2015). Incremental lstm-based dialog state tracker. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015, Scottsdale, AZ, USA, December 13-17, 2015*, pages 757–762.