Designing Coherent And Engaging Open-Domain Conversational AI Systems

by

Ioannis Papaioannou

Submitted for the degree of

Doctor of Philosophy

School of Mathematical and Computer Sciences

Heriot-Watt University

May 2022

"All have their worth and each contributes to the worth of the others."

– J.R.R. Tolkien

Abstract

Designing conversational AI systems able to engage in open-domain 'social' conversation is extremely challenging and a frontier of current research. Such systems are required to have extensive awareness of the dialogue context and world knowledge, the user intents and interests, requiring more complicated language understanding, dialogue management, and state and topic tracking mechanisms compared to traditional task-oriented dialogue systems. Given the wide coverage of topics in open-domain dialogue, the conversation can span multiple turns where a number of complex linguistic phenomena (e.g. ellipsis and anaphora) are present and should be resolved for the system to be contextually aware. Such systems also need to be engaging, keeping the users' interest over long conversations. These are only some of the challenges that open-domain dialogue systems face. Therefore this thesis focuses on designing dialogue systems able to hold extensive open-domain conversations in a coherent, engaging, and appropriate manner over multiple turns.

First, different types of dialogue systems architecture and design decisions are discussed for social open-domain conversations, along with relevant evaluation metrics. A modular architecture for ensemble-based conversational systems is presented, called Alana, a finalist in the Amazon Alexa Prize Challenge in 2017 and 2018, able to tackle many of the challenges for open-domain social conversation. The system combines different features such as topic tracking, contextual Natural Language understanding, entity linking, user modelling, information retrieval, and response ranking, using a rich representation of dialogue state.

The thesis next analyses the performance of the 2017 system and describes the upgrades developed for the 2018 system. This leads to an analysis and comparison of the real-user data collected in both years with different system configurations, allowing assessment of the impact of different design decisions and modules.

Finally, Alana was integrated into an embodied robotic platform and enhanced with the ability to also perform tasks. This system was deployed and evaluated in a shopping mall in Finland. Further analysis of the added embodiment is presented and discussed, as well as the challenges of translating open-domain dialogue systems into other languages. Data analysis of the collected real-user data shows the importance of a variety of features developed and decisions made in the design of the Alana system.

Dedicated to my parents, for without their continuous support, encouragement, and love, none of this would be possible . . .

Acknowledgements

First and foremost, I would like to thank my supervisors, Oliver Lemon and Ioannis Konstas, for their continuous support, guidance, advice, and patience over all these years. Oliver was the person who made me, a person who wouldn't touch a conversational system with a 10ft pole due to heavy stuttering issues, fall in love and pursue this field of science. Both their continuous feedback kept me focused and determined to reach that bright light at the end of the dark, dark tunnel that is a PhD journey.

A huge thanks goes to my team-mates during the Amazon Alexa Prize competitions, Amanda Curry, Jose Part, Igor Shalyminov, Xinnuo Xu, Yanchao Yu, Alessandro Suglia, Shubham Agarwal, and Ondrej Dušek, without which the Alana system would have never seen the light of day... Special thanks also goes to Verena Rieser and Arash Eshghi for their work, constructive feedback, and support throughout these years both during and outside of the Alexa Prize competitions.

Warmest gratitude would also go to Christian Dondrup and Barbara Girard not only for their invaluable advice, but also for their understanding and support during the stressful times. Their constant reminder of what I was fighting for, helped me remain focused and finish this thesis in time... This also goes to Emilyann Nault, for being extremely supportive, being there when I had to vent and always lending her advice and help when requested.

My deepest thanks also go to my brother, sister, and my parents who even being miles away (2,822 to be exact – as a laden swallow flies) their love and support was always there throughout these years.

Research Thesis Submission

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

Name:	Ioannis Papaioannou		
School:	School of Mathematical and Computer Sciences		
Version: *(i.e. First, Resubmission, Final)*	Final	Degree Sought:	Doctor of Philosophy

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:	03/05/2022

Submission

Submitted By *(name in capitals)*:	IOANNIS PAPAIOANNOU
Signature of Individual Submitting:	
Date Submitted:	03/05/2022

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:				

Contents

List of Figures

List of Tables

Acronyms

ASR Automatic Speech Recogniser. xi, 12–15, 28, 39, 47, 48, 82, 83, 89, 108, 109, 118, 132, 133, 139

CA Conversational Agent. 136

DA Dialogue Act. 15, 27, 33, 40, 146

DM Dialogue Manager. 5, 12, 19–21, 26, 27, 29, 39, 41, 50, 66, 140, 155, 158

DNN Deep Neural Network. 12, 14, 28, 29, 41

E2E End-to-End. 12, 13, 29–31, 40, 41

ECA Embodied Conversational Agent. 136

GLA University of Glasgow. 137

HCI Human-Computer Interaction. 2

HHI Human-Human Interaction. 32, 159

HMM Hidden Markov Model. 14, 32

HRI Human-Robot Interaction. 135, 136

HWU Heriot-Watt University. 137, 146

IDIAP Idiap Research Institute. 137

IDP IdeaPark. 138, 150

LAAS LAAS-CNRS. 138, 142

LM Language Model. 43

MDP Markov Decision Process. 21, 25, 113

MSE Mean Square Error. 64

MT Machine Translation. 147

MuMMER Multi-Modal Mall Entertainment Robot. 137, 153

NE Named Entities. xi, 4, 18, 20, 49, 52, 55, 63, 66, 74, 76–79, 82, 83, 95, 96, 98, 100, 102, 104, 107, 126, 127, 155, 156

NER Named Entity Recognition. 95–97

NLG Natural Language Generation. 12, 26–29, 34, 66, 82, 94, 102, 116, 127, 147, 153

NLU Natural Language Understanding. 4, 12, 15, 18, 20, 29, 47, 66, 70, 82, 85, 89, 90, 93, 95, 101, 102, 108, 113, 139, 140, 144, 147, 155, 159

NP Noun Phrases. 49, 52, 55, 63, 100, 104

OOD Out-Of-Domain. 58, 59

POMDP Partially Observable Markov Decision Process. 21, 41

RL Reinforcement Learning. 21, 26, 40, 41, 113, 114, 116, 140, 153

RNN Recurrent Neural Network. 27

ROMULUS Robotics-Oriented MUltitask Language UnderStanding. xiii, 139, 146–148

Seq2Seq Sequence-To-Sequence. x, 27, 28, 30, 31, 59

TTS Text-To-Speech. 12, 22, 28, 29, 39, 139–141, 144, 148

VTT VTT Technical Research Centre of Finland. 138, 150, 153

Chapter 1

Introduction

Humans are able to communicate with each other using various channels, including verbal and non-verbal (gestures, facial expressions, etc) communication forms. Language is one of the most efficient, but complex form of communication we, as humans, have. From day-to-day social interactions, to complicated transactional conversations, we use language to convey all sorts of messages to one or multiple parties. Language, being an extremely powerful tool, has evolved over the thousands of years to be efficient, direct, artistic, and complicated at the same time.

Although linguists have categorised language and communication using different structures and models throughout the years (Austin, 1975, Grice, 1975, Grosz and Sidner, 1986), including the development of theories of conversation based on speech acts and shared plans (Austin, 1975, Grice, 1975), this thesis is mainly focused on the following three categories of dialogue between humans and artificial agents:

1. **task-oriented (or task-based)**: Transactional dialogue where the turns of the conversation aim to complete a set task (e.g. making a restaurant reservation).
2. **domain-specific conversation**: Conversing about a predefined set of topics (e.g. talking about movies, but not able to talk about politics).

3. **open-domain conversation (or social dialogue)**: Conversing on any topic any of the interlocutors wishes.

These categories can also be applied to conversations between humans and artificial systems. The interest and need for a Human-Computer Interaction (HCI) that enables its users to interact with it using language (written or verbal) started as early as the 1960s. Early systems such as SHRDLU Winograd (1972) and GUS Bobrow et al. (1977) aimed at researching and understanding natural language, as well as the design of the first chatbot called ELIZA (Weizenbaum, 1966), designed as a psychotherapy agent, based on reacting to and mirroring the user's utterance back to them (e.g. USER: *"I am feeling sick today"*, ELIZA: *"I am sorry that you feel sick today"*).

Since then, chatbot and spoken dialogue system technology has progressed considerably (see McTear (2020) for a summary), with the introduction of fully voice-enabled personal assistants such as Amazon Alexa, Siri, Google Assistant, and many more. These systems however focus more on task-oriented interaction and single-shot (single user-system turns) task-based dialogues while lacking any form of context, memory, or open-domain capability.

This thesis focuses on how real world open-domain systems engaging in social interaction can be designed, built, and evaluated.

1.1 Challenges of open-domain conversation

Building an artificial dialogue system able to perform open-domain conversation has been an elusive and long-standing aim for artificial intelligence. Early systems such as ELIZA (Weizenbaum, 1966) and A.L.I.C.E. (Wallace, 2009) (despite playing a fundamental role in advancing conversational agents' technology) were able to perform dialogue only in constrained environments. Recent advances in hardware technology, Machine Learning, and Neural Networks, as well as the collection of huge datasets made it possible to develop systems like XiaoIce (Shum et al., 2018, Zhou et al., 2018), Mitsuku[1] and Language Models like GPT-2 (Radford et al., 2019) and GPT-3 Brown et al. (2020b) where users are able to engage in dialogue with artificial dialogue systems in a seemingly open-domain manner, these systems however are still brittle and often provide unreliable responses (Li et al., 2017).

As a number of researchers conclude (Gao et al., 2019, Huang et al., 2019, Levin et al., 2000, Rieser and Lemon, 2008, Young, 2010), *task-oriented* dialogues are easier to optimise since the goal that needs to be reached (e.g. if viewed as an optimal decision making process where the agent is trying to optimise its reward, as will be discussed in Section 2.1.1.3) is easily definable. Similarly, in domain-specific dialogues (as will be discussed in more detail in the following chapters) given enough appropriate in-domain data, a neural model can be trained in order for the agent to hold a conversation with the user with some form of context representation.

On the other hand, *open-domain* dialogue systems are optimised for more abstract goals such being entertaining, being able to provide recommendations, engage in interesting conversation on a specific topic, or providing emotional support. These goals are hard to define since they vary according to the participants, the culture, the situation, etc, creating many different ways in which a concept can be formulated. Furthermore, a system such as this requires to have a much deeper understanding of the user inputs and what has transpired so far in the conversation (dialogue context), pay close attention to the user's needs and intents, and be able to generate responses using a consistent personality (Huang et al., 2019). Furthermore, given the freedom of topic in an open-domain conversation, maintaining a coherent conversation that is able to provide relevant information to what the user enquired (meaning that the information is *grounded* on the user's request and topic (Clark, 1996, Clark and Brennan, 1991)) across multiple conversational turns is also much harder than in a domain-specific or task-based conversation (Ghazvininejad et al., 2018, Qin et al., 2019).

[1] https://www.kuki.ai/research

In recent years, advances in deep learning, access to better hardware, and increased data availability allow the advancement of conversational AI technology and have shown big improvements over the early days systems, however (given the challenging nature of modelling human language) several of the challenges in the field still remain unsolved. In order to implement an artificial agent able to engage in open-domain conversation, several challenges need to be addressed to be able to handle the wide coverage such dialogues require in a coherent but also engaging way detailed as below. Each of these notions will be explained in more detail in the following chapters:

- **Open NLU:** The Natural Language Understanding (NLU) module needs to be general enough, due to the vast amount of possible user intents and entities the system needs to recognise (Sections 2.1.1.2, 5.2.2).
- **Co-reference and Anaphora resolution:** Such dialogue systems may fail to create consistent and coherent responses if referring expressions and information omissions are not resolved (Sections 2.1.1.2, 3.3.2, 2.1.1.3). E.g. - *"The CEO of Microsoft now is* ***Satya Nadella****". - "How old is* ***he****?"*
- **Ellipsis resolution:** In multi-turn conversations the interlocutors often respond to the previous statement using elliptic sentences (e.g. SYS: *"Do you like ice-cream?"*, USR: *"I do"* → *"I like ice-cream"*). Such systems should be able to recognise these and expand the elliptic sentences to put them in context (Sections 2.1.1.2, 3.3.2, 5.2.1).
- **Named Entities (NE) understanding:** It is not always enough to just recognise NE, but understand the links between them and how they impact the context of the conversation or the underlying topic (Sections 5.2.1, 5.2.2, 5.2.3.4).
- **Topic tracking:** In a natural flow of an open-domain conversation, the topic rarely remains the same. The interlocutors jump from topic to topic in a coherent way, even backwards to previously mentioned topics (Section 5.1.1).

- **Ethics/Abuse:** Allowing the user (and the system) to engage freely on any subject, increases the probability of potential unethical or abusive content to be delivered on either direction (Sections 5.2.5, 5.2.3.7). This needs to be detected and properly handled by the system.

- **Personality:** In order to remain engaging, the system needs to have a personality of its own. This enables more natural conversation on topics that are interesting to the users (i.e. SYS: *"What is your favourite book? Mine is The Lord of the Rings."* (Section 3.3.3))

- **Question Answering:** An open-domain conversational system needs to be able to answer all sorts of questions the user might ask. This enables natural follow-up turns and a better flow of conversation (Sections 3.3.3.3, 5.2.3.3, 5.2.3.2, 5.2.3.4).

- **Dialogue Management:** Given the non-specific goal nature of open-domain conversation, the way the Dialogue Manager (DM) decides how it should respond on each turn becomes increasingly challenging (Sections 5.3, 3.3.4, 3.4).

- **Discover user interests - user modelling:** To keep the user interested and engaged, the system needs to first understand what topics the user is interested in or which topics the user considers boring or dislikes (Section 5.2.4). E.g. *U: "I don't like sports.", S: "I've heard this interesting article about sports today. ..." (Negative example or user intent interpretation.)*

- **Clarification:** When having access to multiple data-sources and vast amounts of information, quite often the entities mentioned are ambiguous (e.g. talking about *Angels and Demons* which exists as both a movie and a book). Processes need to be implemented to disambiguate such entities and ground them to the current topic (Section 5.2.3.5).

- **Chit-chat:** An open-domain conversation includes a lot of social language which makes the conversation flow naturally (Section 5.2.3.8, 3.3.3.1, 3.3.3.2).

- **Sentiment:** As in a conversation between humans, the system needs to be aware of the sentiment the user's utterances have. This signal would allow the system to decide its actions more appropriately (Section 3.4.1, 3.3.3.9).

- **Response diversity:** Given the open-ended style of conversation with such dialogue systems, it is important that the responses generated by the system have enough variance in terms of style, length, and vocabulary, as well as a wide topical coverage. This will enable such systems to be more engaging and enjoyable to interact with (Sections 3.3.3).

- **Persona consistency:** During the chit-chat turns throughout the dialogue the system should present a consistent personality. This allows for better flow of coherent conversation when the system does not contradict itself (Section 3.3.3.1). E.g. *"S: My dog's name is Estelle."* then later during the same conversation *"U: How old is your dog?", "S: I don't have a dog".*

- **Safety:** Several user enquiries might require special handling. This is particularly important when utterances involving the user's or others safety are detected (e.g. *"I want to kill myself"* or *"How can I hide a dead body?"*). These need to be detected and carefully constructed responses should be provided, meaning that an appropriate level of control in the system's responses needs to be maintained (Section 3.3.3.1).

- **Evaluation:** Given the complexity of such systems, proper evaluation metrics need to be considered that cover all the different aspects of an open-domain dialogue setting (Sections 2.2, Chapters 4 and 6).

Although these challenges have been tackled individually or in subsets within the academic community, this thesis focuses on how to design, build, and evaluate such a complete system that addresses all of the above issues.

1.2 Project Statement and Research Questions

In this thesis, the aforementioned challenges will be addressed by describing and evaluating an open-domain conversational AI system designed, implemented, and deployed during the Amazon Alexa Prize 2017 and 2018 (Chapters 3 and 5 respectively) that was designed to be able to engage in coherent and engaging conversation over multiple turns.

The research questions discussed in this thesis are the following:

1. **How can a Conversational AI system be designed to tackle all of the challenges mentioned in Section 1.1?**
2. **How can such a system be optimised to hold engaging and coherent conversation with its users?**
3. **How can such a system be properly evaluated?**
4. **How scalable can such a system be when deployed in large populations and what are some of the engineering challenges to be solved?**
5. **How does an additional embodiment (using a robot) and the ability to also perform tasks affect the user's perception and acceptance of an open-domain conversational system in a real-life scenario?**

1.3 Objectives

Through the work presented in this thesis, the following contributions have been made:

1. Describe an **architectural design** and **methodology** for building an open-domain conversational AI system with enhanced context modelling through

explicit state representation and user modelling. The architecture is based on an ensemble of different bots competing for selection by a trained ranker (Chapters 3, 5).

2. Techniques for **improved topical coherence and engagement** in open-domain settings, allowing the conversation to move forwards in a more natural way (Chapters 3, 5).

3. Quantitative **evaluation** of the proposed architecture and analysis of its individual components (Chapters 4, 6).

4. Extension of the proposed open-domain conversational system with **task-performing** capabilities (Chapter 7).

5. Integration of the proposed open-domain conversational system into a robotic agent and analysis of the added **embodiment**'s effect on the system's perception by the users (Chapter 7).

1.4 Chapter Outline

In Chapter 1 the core ideas and differences between an open-domain dialogue and task-oriented or domain-specific dialogues are introduced, as well as the challenges that need to be considered while designing such systems.

In Chapter 2 related work in the field is explored, focused primarily on the different types of conversational AI system architectures available, and their capacity to create open-domain agents. Additionally, different metrics for dialogue system evaluation are discussed.

In Chapter 3, Alana, an open-domain conversational AI system that competed in the Amazon Alexa Prize 2017 is described in detail, and how the challenges listed in Chapter 1 are addressed.

Following that, Chapter 4 describes the motivation, methods, and findings of the data analysis of the data collected during the 2017 competition, aiming to find features that improve the user ratings in open-domain settings.

In Chapter 5 an updated version of the Alana system is described, competing in the 2018 Amazon Alexa Prize. Details of the improvements and additions made to the system and their motivations are further discussed.

In Chapter 6 a similar data analysis to Chapter 4 is described, discussing the changes, additions, and improvements of the 2018 system version over the 2017 one.

Chapter 7 presents a real use-case application of Alana, integrated into an embodied robotic system and deployed in a shopping mall in Finland as part of the MuMMER project. Further analysis of the impact of embodiment in open-domain interaction is explored.

Finally, Chapter 8 summarises and reflects on the work presented in the this thesis, and further possible future work is discussed.

1.5 List of Publications related to thesis

Papaioannou Ioannis, Amanda Cercas Curry, Jose L Part, Igor Shalyminov, Xinnuo Xu, Yanchao Yu, Ondrej Dušek, Verena Rieser, and Oliver Lemon. Alana: Social dialogueusing an ensemble model and a ranker trained on user feedback. *Proc. AWS re:INVENT,* 2017a

Papaioannou Ioannis, Amanda Cercas Curry, Jose L Part, Igor Shalyminov, Xinnuo Xu, Yanchao Yu, Ondrej Dušek, Verena Rieser, and Oliver Lemon. An ensemble model with ranking for social dialogue. *arXiv:1712.07558,* 2017b.

Papaioannou Ioannis, Christian Dondrup, Jekaterina Novikova, and Oliver Lemon. Hybrid chat and task dialogue for more engaging hri using reinforcement learning. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 593–598. IEEE,* 2017c.

Papaioannou Ioannis, and Oliver Lemon. Combining chat and task-based multimodal dialogue for more engaging hri: A scalable method using reinforcement learning. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, pages 365–366,* 2017

Christian Dondrup, **Papaioannou, Ioannis**, Jekaterina Novikova, and Oliver Lemon. Introducing a ROS based planning and execution framework for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI International Workshop on Investigating Social Interactions with Artificial Agents,* ISIAA 2017, pages 27–28, 2017. doi: 10.1145/3139491.3139500. URL http://doi.acm.org/10.1145/3139491.3139500.

Novikova, Jekaterina, Christian Dondrup, **Ioannis Papaioannou** and Oliver Lemon. Sympathy Begins with a Smile, Intelligence Begins with a Word: Use of Multimodal Features in Spoken Human-Robot Interaction.In *RoboNLP@ACL*, 2017.

Amanda Cercas Curry, **Papaioannou Ioannis**, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xinnuo Xu, Ondrej Dušek, Arash Eshghi, Ioannis Konstas, Verena Rieser, et al. Alana v2: Entertaining and informative open-domain social dialogue using ontologies and entity linking. In *1st Proceedings of Alexa Prize (Alexa Prize 2018),* 2018.

Papaioannou Ioannis, Christian Dondrup, and Oliver Lemon. Human-robot interaction requires more than slot filling - multi-threaded dialogue for collaborative tasks and social conversation. In *FAIM/ISCA Workshop on Artificial Intelligence for Multimodal Human Robot Interaction,* pages 61–64, 2018.

Christian Dondrup, **Papaioannou Ioannis**, and Oliver Lemon. Petri net machines for human-agent interaction. *arXiv:1909.06174*, 2019.

Mary Ellen Foster, Bart Craenen, Amol Deshmukh, Oliver Lemon, Emanuele Bastianelli, Christian Dondrup, **Papaioannou Ioannis**, Andrea Vanzo, Jean-Marc Odobez, Olivier Canévet, et al. Mummer: Socially intelligent human-robot interaction in public spaces. *arXiv, abs/1909.06749*, 2019.

Chapter 2

Literature Review: Concepts and background in Spoken Dialogue Systems

In this chapter, the architectures and components of typical spoken dialogue systems will be reviewed, both from a task-oriented and open-domain dialogue perspective, and the challenges for these components in open-domain conversations will be discussed.

Additionally, common structures, architectures, and methods for designing open-domain and task-oriented dialogue systems will be explored.

Finally, various different metrics for the evaluation of the performance of spoken dialogue systems will be presented.

2.1 Spoken Dialogue Systems Architectures

Dialogue systems can be designed to interact with users in a multitude of ways.

Traditional dialogue systems include text-based (where the user types in their queries as text) and speech-based (the user speaks directly to the agent with the use of a microphone) interaction with the users. There can be additional inputs to a dialogue system such as visual information or other forms of signals, in which case the agent becomes *multi-modal*. A use-case of such a system is described in Chapter 7.

Conversational systems can be divided roughly into three main categories in terms of their architecture: *modular* architecture, *End-to-End (E2E)* architecture, and *ensemble* architecture.

In a **modular architecture**, the system consists of a number of different components (ASR, NLU, DM, Natural Language Generation (NLG), Text-To-Speech (TTS)) creating a pipeline from the user's utterance (input) until a response has been returned to the user (output). Most of these modules can be implemented using hand-crafted or statistically trained models, as well as data-driven approaches learning directly from data, for exampled by employing Deep Neural Networks (DNNs). Data-driven (or neural) modules can utilise the underlying information that can be extracted from data of available dialogues, effectively trying to interpret the twists, turns, and nuances of expression existing in human language.

On the other hand, **E2E** architectures are able to substitute some of those components with a DNN which has been trained on a large number of human conversations, drastically reducing the amount of manual work required to design and implement those individual components.

In an **ensemble architecture** a more hybrid approach is followed, developing different components for different topics (e.g. news, weather, etc) combining different response generation approaches such as neural, information retrieval (Banchs and Li, 2012), or template-based into an ensemble of responses, then applying a ranking function to select the appropriate response.

From the point of view of open-domain conversation, all of these approaches have advantages and disadvantages which will be discussed in more detail below.

2.1.1 Modular Architecture

This type of architecture was the predominant paradigm in the 1990s (up until the introduction of neural E2E in $\sim$ 2015). It was used in a variety of task-oriented dialogue systems, such as the COMMUNICATOR project systems (Walker et al., 2001a,b, 2002) in industry as well as academic research. A typical task-oriented dialogue system follows the architecture outlined in figure 2.1 and consists of the following components.

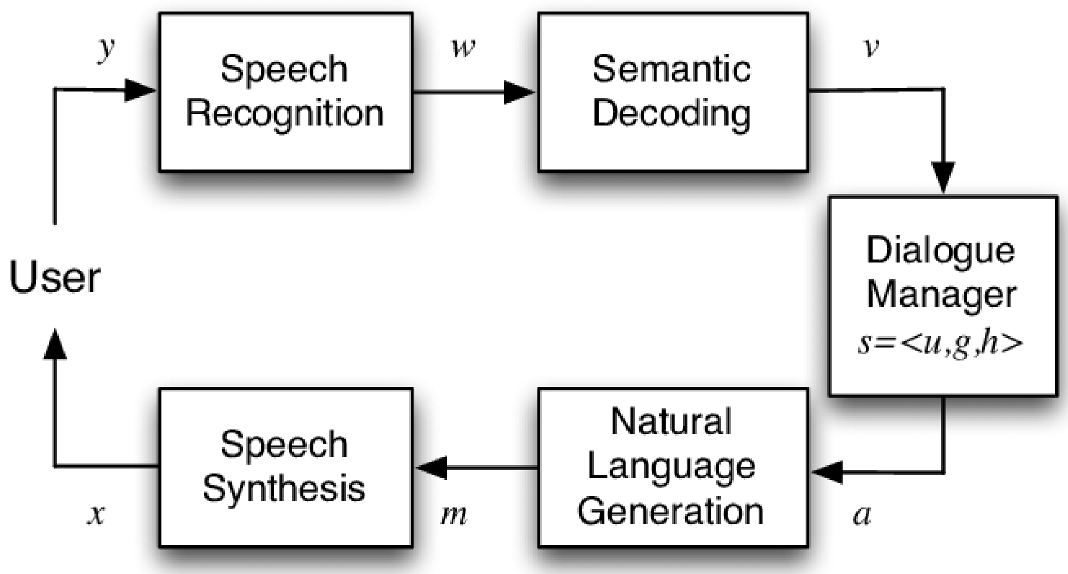


FIGURE 2.1: Typical architecture of a Spoken Dialogue System as presented by Young (2010)

Below, the different components of a typical modular architecture are described.

2.1.1.1 Automatic Speech Recogniser

This module is required when a speech interface is present in a dialogue system, enabling the user to use their voice as input to the system. As Young (2010) describes, the Speech Recognition module is taking an acoustic signal of the user's utterance y and outputs a sequence of words w. This sequence is typically a distribution over the different word hypotheses (usually presented in the form of an N-best list (Ostendorf et al., 1991)) of the string representation of the user's utterance $p(w|y, \lambda_{asr})$, where λ_{asr} are the configuration parameters of the ASR.

$p(w, y)$ is then decomposed into an acoustic model and a language model. The acoustic model is usually trained to understand a set number of words $w \in V$ with V consisting of the module's vocabulary, usually words that are relevant to a specific domain (e.g. a dialogue system responsible for booking a table at a restaurant should be able to recognise all relevant words used in such a conversation, e.g. days, time, numerical values, etc). In open-domain conversations however, the vocabulary V needs to encompass a large breadth of domains and topics, increasing the ASR's footprint by a margin.

Presently, two main categories of ASR systems exist: hybrid (Bourlard and Morgan, 2012) and end-to-end (Graves, 2012). Hybrid ASR combines Hidden Markov Model (HMM) with DNN models to train three independent components: an acoustic model (which estimates the posterior probabilities of the HMM states), a language model (which estimates the probability of a specific word appearing in a sequence), and a pronunciation model (which maps phonemes to specific words). On the other hand, an end-to-end system learns to directly map sounds to specific words. Both of these technologies come with a different set of challenges (Jain et al., 2020) when dealing with similarly-sounding words, especially in an open-domain set-up. For example, *"It's easy to recognise speech"* and *"It's easy to wreck a nice beach"* might sound very similar, but carry completely different meaning. Thus, ASR technologies would ideally be context sensitive to facilitate open-domain Spoken Dialogue Systems.

In more recent years, with the development of cloud-based, large-vocabulary ASRs, the performance of these modules has improved drastically, and they are now able to even distinguish between different accents (e.g. Google Speech-To-Text[1], Amazon Transcribe[2], Microsoft ASR[3]).

[1]https://cloud.google.com/speech-to-text
[2]https://aws.amazon.com/transcribe/
[3]https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/

2.1.1.2 Natural Language Understanding

After the ASR, the NLU module (also known as semantic decoder) tries to parse a word sequence (usually the top-scored result from the ASR) into a meaning representation that the system can understand. Given the NLU's parameter's λ_{nlu} and it's internal model $p(u|w, \lambda_{nlu})$, this module produces a semantic representation u of the user's utterance y as:

$$p(u|y, \lambda_{asr}, \lambda_{nlu}) = \sum_{w} p(u|w, \lambda_{nlu}) p(w|y, \lambda_{asr}) \tag{2.1}$$

As an example, let's assume that the user's utterance is *"I would like to have a cheeseburger and a large coke"*. For the task-oriented system to understand what the user is enquiring for (also known as a user *intent*), it uses an abstract representation of the utterance called Dialogue Act (DA), for example, as follows:

$$DA = order : \{food : cheeseburger, drink : \{type : coke, size : large\}\}$$

Many DAs schemes have been proposed, such as DATE (Walker and Passonneau, 2001) and DAMSL (Allen and Core, 1997). DAs are supposed to be domain-general whereas intents are domain-specific. DAs have been widely used in task-oriented systems in the past, and user intents (the *intent* in this example being `order` with intent-specific *values* of `cheeseburger` and `large coke`) can be classified and represented in various ways. Traditionally, these intents are captured using hand-crafted rules (e.g. using pattern-matching techniques, such us regular expressions or scripting languages like AIML (Marietto et al., 2013)) or grammars, modelling the various ways such a request can be uttered.

The way intents are used by the NLU module needs to be carefully designed and is usually domain-specific. E.g. a restaurant conversational system might have acts (and vocabulary) around `order`, `menu`, `request`, `cancel_order`, etc, while a hospital conversational system might have `admit`, `attend`, etc.

An additional challenge to be addressed is the significance of context in a given utterance, which can be used to resolve *syntactically ambiguous* sentences. Consider for example the sentences *"Enraged Cow Injures Farmer With Axe", "Stolen Painting Found by Tree"*, or *"Kids Make Nutritious Snacks"*, where their meaning can be interpreted differently depending on how the sentences are read. Although they can be easily understood by a human (i.e. An ax carrying farmer got injured by an enraged cow) since the context in this case is implicit, a conversational agent needs additional information to make this distinction.

Additionally, words might carry a different meaning depending on the context they are presented in Devlin et al. (2018). For example, in the sentences *"It was too much to **bear**"* and *"As soon as he saw the **bear** he fainted"*, the word "bear" means different things based on the surrounding sentence. Likewise, an entity (e.g. The Lord of the Rings) can be ambiguous as it can be associated with different properties depending on the context it's been discussed in (e.g. books or movies).

These challenges are prominent in open-domain environments, where the number of intents and different contexts is infinite making modeling the meaning of the user's language an extremely challenging task.

Rule-based Language Understanding and intent recognition can be exponentially expensive when more domains with wider intent coverage are added to the system's, since rules need to be hand-crafted by domain experts. This is something that can be tackled with using a neural approach.

In **neural architectures**, the understanding of the user's query is not based on some hand-engineered representations. To extract the meaning of the user's utterance, the input words are represented in a numerical format called *word embeddings* (Bengio et al., 2003, Mikolov et al., 2013c). Word embeddings, in short, are functions that map words to high-dimensional vectors, where each value in the vector can represent different features related to the word, that are able to also encode semantic or grammatical interpretation (Turian et al., 2010). The notion of word embeddings stems from the Distributional Hypothesis according to which, words that appear in similar linguistic contexts are likely to have related meanings (Harris, 1954). By encoding words using numerical high-dimensional vectors, a model fed numerous examples of sentences, such as:

- "The restaurant we went to last night was great"
- "I believe that restaurant closes at 23:00"
- "Will they serve pizza in that restaurant"
- "Forget the movies. Let's go eat. There is a restaurant nearby"

might not be able to grasp the concept of a *restaurant* (as in a building where people go to eat), it will however be able to understand, though its semantic meaning, what this word means in a given context. This is based on the *co-occurrence* of the terms in the examples in the text corpus, meaning the frequency of any bi-gram in the sentence to appear in other sentences in the same order (Bordag, 2008). Word co-occurrence is often used to calculate how similar the *meaning* between two texts is, which is also known as *semantic similarity.*

Additionally, words with similar semantic meaning tend to have higher *cosine similarity* than others in the vector space, forming clusters of words with similar context (e.g. *"garden"*, *"sprinkler"*, and *"hose"* will be closer than *"oven"* or *"refrigerator"* (Fig. 2.2)).

In recent years, a number of different word embedding models have been researched in order to increase the contextual information encoded in them, such as GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013a). With the introduction of the transformers model (Vaswani et al., 2017) and the widely used BERT model (Devlin et al., 2018), a new, more contextually rich embeddings architecture arose, leading to a much better generalisation of the user intents.

The wide embrace of transformer-based architectures brought in an influx of similar models that stem from BERT attempting to optimise and further improve the encoding of context in dialogue systems, but also reduce the computational power required to train such models. Of these, most notable would be the RoBERTa

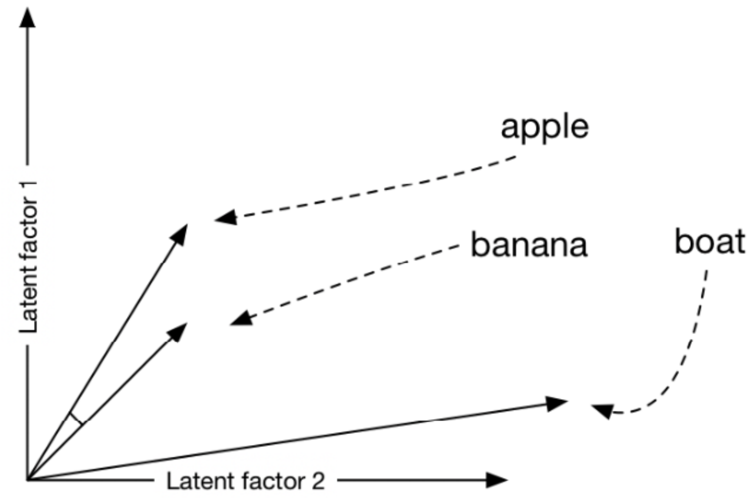


FIGURE 2.2: Simple representation of a vector space. Vectors of words with similar semantic meaning (apple, banana) tend to appear closer together than others (boat). [Image taken from `https://erikbern.com/2015/09/24/nearest-neighbor-methods-vector-models-part-1.html`]

model (Liu et al., 2019) which improves over its predecessor in terms of speed and computational power required. Beltagy et al. (2020) addressed the limitation of the transformer-based architecture with regards to its input size to the model by introducing the Longformer model, allowing even whole dialogues to be encoded in a single model run.

In open-domain conversation in particular, the NLU needs to not only identify and understand topics and Named Entities (NE), but it ideally should also put them in the right context, potentially resolving ambiguities, or inferring the topic from a given NE.

In the work on the conversational system presented in Chapters 3 and 5, a mixture of these different techniques is used to capture the user's intent, including pattern-matching (using regular expressions) as well as trained neural models using GloVe and Word2Vec word embeddings.

2.1.1.3 Dialogue Manager

The DM performs two main tasks: it is the module responsible for maintaining the context (usually through some form of Dialogue State Tracking), and choosing the action the system should take in each consequent turn. In **task-oriented** systems the DM usually has a goal to reach, which could be for example to complete a reservation in a restaurant or book a plane ticket. This can be approached again either by using *rule-based* policies, or following *statistical* approaches.

A commonly used method of modelling tasks using hand-crafted rules is the *slot-filling* technique. In this, the system keeps an internal structure of the properties of the task as empty fields (or slots), where it elicits responses from the user filling this information. For example, a hotel reservation conversational system might have the following representation for booking a room:

```
room_booking:
    date_in:
    date_out:
    no_occupiers:
    [
        occupier:
            name:
            surname:
            age:
    ]
```

In this example, for the goal (room booking) to be successfully completed the system requires all the fields to be filled which can be provided by the user either in a single turn, or multiple subsequent turns. If the information is not provided, the system needs to have mechanisms to elicit this information from the user. This requires the DM to keep track of the history h of the conversation so far, along with keeping track of the information provided using some form of *conversation state* representation $s_t =< u, g, h >$ (Young, 2010) given a user goal g and a dialogue act u. In other words, the dialogue state at time t is an abstract representation of all turns in the dialogue until $t-1$ (Zhang et al., 2020).

State representations can be implemented in a variety of other ways. In the context of *multi-turn* conversation though that usually follows an open-domain conversation, it is paramount that the representation is able to track information across turns and put those in the right context on each turn. For example, if the user's utterance is *"five"* following a system's enquiry *"how many people will attend?"* (an *elliptic* utterance), it needs to be *resolved* in context with that system's turn.

Ellipsis resolution (Johnson, 2001) is the linguistic phenomenon which puts an elliptic (or fragmented) sentence into the right context based on the history of the conversation. In the example above, in a contextual NLU, *"five"* would be resolved into *"five people will attend"*.

Another linguistic phenomenon frequently present in both task-oriented and open-domain systems when the user is engaged in a multi-turn conversation, is *anaphora resolution* (Mitkov, 1999), which simply put is the problem of resolving what a Noun or Pronoun phrase refers to (usually a NE) taking into account the correct gender. For example in the turn *"- I saw Mary yesterday", "- How was she?" she* refers to *Mary*. Again, this requires a contextual NLU, which requires a dialogue state for resolution.

Generally, context needs to be represented in a way that can be shared across turns and *impact* the way the DM decides. This is even more important in an open-domain setting, where the context might include a wide range of properties like the current *topic* discussed, specific NEs, etc.

Another task of the DM includes some underlying logic on what the system should do in cases where the user's utterance does not fit any of the pre-defined intents that the system is able to handle, or in task-oriented and domain-specific architectures, how to handle those out-of-domain cases. The simplest policy in those situations would be to just notify the user that this exceeds the agent's capabilities (e.g. *"I am sorry I am not sure what you mean"*) which has been traditionally employed in a variety of task-oriented systems (Shum et al., 2018, Zhou et al., 2018). In open-domain settings, this phenomenon is much more frequent than in task-oriented dialogue, given the wide breadth of topics the user should be able to talk about, so more sophisticated methods for fallback strategies are needed. For example, a conversational system would be taking initiative and asking the user a question to progress the conversation.

As seen so far, in a **hand-crafted** system architecture, in order for the DM to decide which action it should take on the next turn, the designer should take into account all the different routes and actions the system would be able to address. Although this procedure can provide fast implementation in small systems and closed-domain applications, it comes with a heavy cost in terms of manual labour required. Additionally, it is error-prone, as it is impossible to anticipate all possible branches a dialogue could take.

An alternative method entails **statistical** modeling techniques (Levin et al., 2000), (Rieser and Lemon, 2011) using Reinforcement Learning (RL), usually by representing each possible dialogue state within a Markov Decision Process (MDP) or a Partially Observable Markov Decision Process (POMDP) (Young et al., 2013b). Then the action the DM needs to take is decided by a *policy*, trying to maximize its expected *utility* (explained below). This solution provides scalable, context-aware systems that provide optimal strategies. On the downside, the system needs to learn from experience, meaning it needs to be fed a lot of domain-specific sample conversations.

In the following, the key concepts of these techniques are described.

Markov Decision Process: MDPs represent the (dialogue) states as ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{T}$, $\mathcal{R}$), where $\mathcal{S}$ represents the state space, consisting of all valid states the system can be in. $\mathcal{A}$ denotes all the possible actions, while $\mathcal{T}$ is the transition function for an action $a \in \mathcal{A}$ from state $s \in \mathcal{S}$ to reach state $s' \in \mathcal{S}$, with a probability of $\mathcal{P}(s'|s, a)$. r is the reward received by a reward function $\mathcal{R}$ that the agent gains if action a is taken in state s. The optimal action at each step is then defined as the action that is able to maximise the expected long-term accumulation of rewards (utility) from a current state onwards (Rieser and Lemon, 2011).

Reward function: When an agent takes an action a to transition from a state s to the next state s' it receives a reward r. This reward is a numerical signal which can be either positive (reward) or negative (penalty), expressing how good the action taken was. Designing a proper reward function that will be able to assign rewards to an action is extremely important in the policy optimisation task (Young et al., 2013b). In task-oriented systems for example, where the agent's role is to complete tasks efficiently, the reward function would be tied to features such as task completions, dialogue length, or user ratings (e.g. using the PARADISE framework described later in Section 2.2).

In many cases a simulation is used to generate the needed interactions to effectively train the transition model. A widely applicable method to collect this data is using a *Wizard-of-Oz* (Rieser and Lemon, 2008, Williams and Young, 2003), imitating human-human interaction, where the "Wizard" is a human using either TTS or voice modulation, fooling the other participant into thinking they are talking to an artificial system. Using the data collected that way, an *optimal policy* can be found by extrapolating data from these interactions.

Value Function: A value function determines the desirability of a particular state for the agent to be in. The value of a given state s under a policy π is the expected utility (the accumulated discounted reward) if the agent starts from state s and takes actions according to policy π, as shown in Equation 2.2.

$$V^\pi(s) = \mathbb{E}_\pi[R_t|s_t = s] \tag{2.2}$$

Optimal Policy: A policy π dictates the system action that can either be represented by mapping the states to actions in a deterministic or stochastic manner (Young et al., 2013a). The *Bellman equation* (Bellman, 1958) (Equation 2.3) is used, in order to converge to an *optimal policy* that maximizes the discounted accumulated rewards gained by following that policy π in a state s,

$$V^*(s) = \max_a [r(s,a) + \gamma \sum_{s'} P(s,a,s')V^*(s')] \tag{2.3}$$

where $V^*(s)$ represents the maximized V^π in a state s, r is the reward gained from performing action a in state s, $P(s,a,s')$ is the probability of taking action a in state s leading to state s', and $\gamma \sum_{s'} P(s,a,s')V^*(s')$ is the sum of said probabilities of expected future values V^*, discounted by a factor γ.

Q-Learning: As described earlier, given a state space S and a possible action space in those states A, the agent has to learn the value of each action in those states. A popular training method is *Q-learning*, where the *value* of a given *state-action* pair is called a *q-value*. Initially those values are set to either an arbitrary fixed value, or set to random values depending on the design, then the agent starts exploring the action-state space. After an action is taken in a state, an observation of the current environment is made, evaluating the outcome. If it leads to an unwanted outcome (meaning the agent got punished or received no reward), the q-value of that action in that state is lowered, increasing the probability of other actions with higher q-value to be selected on the next iteration. Similarly, if the agent is rewarded taking an action in a state, the q-value is increased, making the selection of that action more likely to take place the next time the agent is in that state. Then the q-value is updated according to:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_t + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)] \tag{2.4}$$

Where $Q(s_t,a_t)$ denotes the old value (at the current time step), α is a learning rate factor controlling how fast new values overwrite the old ones (usually, a very high learning rate is initially set, which then gets progressively lowered (*decayed*) (Even-Dar, 2001)), R_{t+1} is the expected *reward* in the next time step (according to previous knowledge), and $\max_a Q(s_{t+1}, a)$ is the expected maximum *q-value* after taking action a.

It should be noted that in Q-Learning, the q-value is updated on the *previous* state-action pair, since the agent must first try the action in order to evaluate it.

In order to select an action in Q-learning, a *policy* is followed. While Sutton (1998) argues that the simplest way to select an action would be to simply select the action A_t with the highest q-value at each time step t, such as $Q_t(A_t) = max_a Q_t(a)$ exploiting the knowledge the agent has gained up to that moment, sometimes is sub-optimal, since it does not allow for much *exploration* of the state space. Instead, we can allow the agent to select the action with the best q-value *most of the time* (with probability $1 - \epsilon$), but also to have a small probability ϵ to select another action applicable in that state *randomly* with equal probability. This policy is called *ϵ-greedy*, and it allows the *exploitation* of past knowledge while at the same time allowing some *exploration*. In order to maximize the *exploration vs exploitation* output, usually the ϵ starts from a high value, and gradually decays as the algorithm converges to the optimal q-values.

Partially Observable Markov Decision Process: To fully implement an MDP solution, the dialogue state must be fully known, or *observable*. As this is often not the case (for example the user's goals could be part of the hidden or "unobserved" state), an extended framework of the MDP is used, called *Partially Observable Markov Decision Process* (POMDP). In POMDPs (Figure 2.3), the possible solutions are notated as $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \mathcal{Z})$, were as before $\mathcal{S}$ represents the state space, $\mathcal{A}$ the action space, $\mathcal{T}$ the transition probability and $\mathcal{R}$ the reward function. As the state now is not fully observable, $\mathcal{O}$ denotes the a set of possible observations by the system, according to its current knowledge, and $\mathcal{Z}$ is the probability $\mathcal{P}(o|s)$ of observing o in state s (Young et al., 2013b). The system's knowledge at any given time forms a *belief state*, which is the probability distribution over all possible states (Lison, 2015), which is constantly updated as the system's knowledge expands, using:

$$b'(s) = \mathcal{P}(s'|a,o) = \eta\mathcal{P}(o|s)\sum_{s\ni\mathcal{S}}\mathcal{P}(s'|s,a)b(s) \tag{2.5}$$

where b is the belief state, a is the action followed by observation o and η is a normalisation factor. The POMDP policy is then formulated by "mapping each possible belief state to its optimal action" (Young et al., 2013b).

Young et al. (2013a) describes the observation probability function as a stochastic model $\mathcal{M}$, while the decided action of each turn is the result of a second stochastic model $\mathcal{P}$. On each turn of the dialogue, a reward is given based on a reward function $\mathcal{R}$. During training, the dialogue model $\mathcal{M}$ and the policy model $\mathcal{P}$ are trying to maximize the accumulative sum of these rewards.

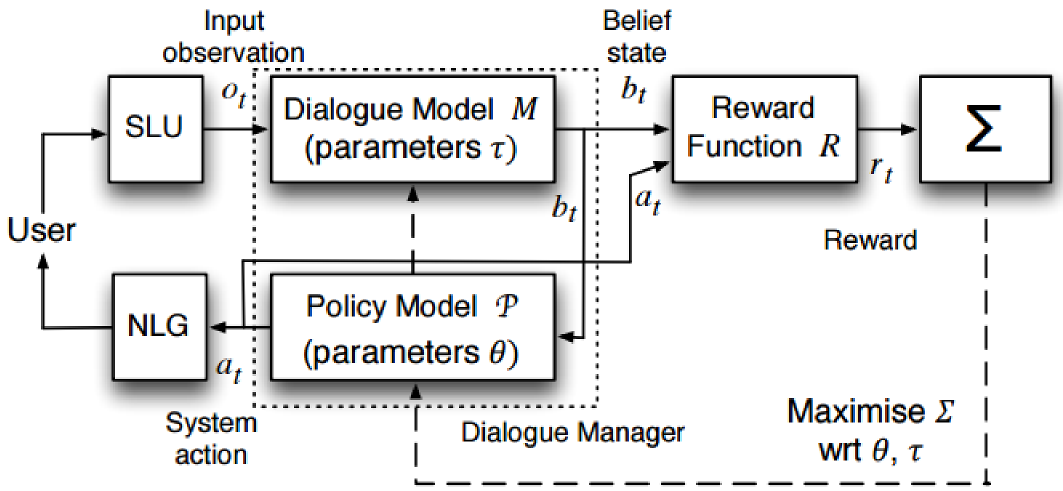


FIGURE 2.3: POMDP components, as shown in Young et al. (2013a)

Similarly to the MDP, a policy π again dictates the system action. The *Bellman equation* (Equation 2.6) is again used but this time over the belief state b_t, in order to converge to the *optimal policy*.

$$V^*(b_t) = \max_{a_t}[r(b_t,a_t) + \gamma\sum_{o_{t+1}}P(o_{t+1}|b_t,a_t)V^*(b_t+1)] \tag{2.6}$$

where $V^*(b_t)$ represents the maximized V^π in a belief state b_t, r is the reward gained from performing action a in belief state b_t and $\gamma \sum_{o_{t+1}} P(o_{t+1}|b_t, a_t)$ is the sum of probabilities of an action a_t within a belief state b_t based on the updated observation model o_{t+1}, discounted by a factor γ.

Henderson et al. (2008) applied these methods to implement a RL system using the data collected in the COMMUNICATOR project (Walker et al., 2001b) as training data instead of policy exploration. DM policy optimisation using RL, however, faces some added challenges, especially in an open-domain setup. As noted by Young et al. (2013b), selecting a proper reward function is key, but often very unreliable when this is extracted directly from users, since their feedback is always objective and hard to be tied to individual features or actions rewarded/penalised. In open-domain dialogue system in particular, this is quite problematic since there is no clear definition of what the underlying task is to use metrics such as task completion (as will be explained in Section 2.2).

In the Alana system described in this thesis, the functionality of the DM is a combination of a ranking function (Section 3.4) and a hand-crafted priority of the generated responses (Section 3.3.4). Additionally, a RL (Q-Learning) policy is used to fine-tune a subset of the generated responses aiming to keep the user engaged.

2.1.1.4 Natural Language Generation

The NLG component, in short, is responsible for taking the action selected by the DM and translating it into meaningful sentences. One of the most widespread approaches is to generate appropriate responses to user requests using template-based generation, where the linguistic structure may contain gaps, producing well-structured results once those gaps are filled (Deemter et al., 2005). This method follows the slot-filling representation as discussed in the previous section and generates the responses using information present in those slots.

A simple example of a template-based system, formulated like the example in Reiter and Dale (1997), could be a semantic representation saying that the movie "Star Wars" starts at 21:00 in screen room 4. Then the DM could produce the DAs of the form $inform_{movie}(title, time, room)$ as:

$$DA = inform_{movie}(StarWars, 2100, 4)$$

Which would then trigger the associated template for NLG:

The movie [title] starts at [time] in screen room [room]

where the gaps (also known as *slots*) *title, time*, and *room* could be filled with information from the DA. This simple technique can be paired with additional technologies to generate context to fill those slots, such as information retrieval response selection (Banchs and Li, 2012). This approach was also used in the Alana conversational system which is further described in Chapters 3 and 5.

Templated rule-based approaches provide a fast and fully controllable option in designing the responses the NLG component will produce. Nonetheless, this approach is fallible to generating a small (and often boring) amount of possible responses leading to low response diversity, as outlined in Section 1.1.

An alternative implementation to designing templates manually and/or to use information retrieved from some knowledge base verbatim, is instead to train a model to generate such responses automatically using previous conversations as training data. The Seq2Seq model architecture (Sutskever et al., 2014) can be used for NLG, based on the Recurrent Neural Network (RNN) architecture. In Seq2Seq, a provided conversation context c is fed to the network (encoder) token-by-token, producing an internal hidden state representation h_t. At each time-step t, the next token in the sequence is fed to the encoder alongside the hidden state so far h_{t-1}, eventually producing a single vector representation encoding all the information in the context. Then another RNN network (decoder) uses this final representation to generate a response token-by-token in a similar but reverse fashion (Figure 2.4).

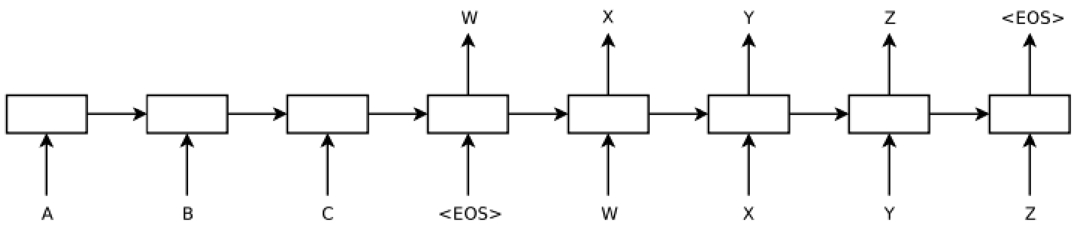


FIGURE 2.4: The architecture of a Seq2Seq model, as presented in Sutskever et al. (2014)

As Sutskever et al. (2014) also state, Seq2Seq models are unaffected by the variation of the input and output sizes, overcoming a challenge traditional DNNs face where the dimensions of the input and output vectors need to remain fixed and known. Although these models provide better response language variation than the rule-based approach, their responses can be boring, inappropriate, and inconsistent (Dušek et al., 2020, Li et al., 2016c, McTear, 2020, Papaioannou et al., 2017a). More recently, a number of transformer-based models are able to produce very fluent output (e.g. GPT-2/3 (Brown et al., 2020b, Radford et al., 2019), BART (Lewis et al., 2019), etc), but the output is still uncontrollable.

In open-domain NLG a mixture of these techniques might be required to facilitate the breadth of variation in possible system outputs. For example on the topic of movies, a template-based approach could be followed, but if the topic shifts to recent news, an information retrieval approach might be optimal, all while interweaving out-of-domain, chit-chat style turns.

2.1.1.5 Text-To-Speech

The TTS module has the opposite functionality of the ASR. It takes the textual response generated by the NLG and using a synthesizer outputs the response using a voice interface to the user.

Although this module is not relevant to the work presented in this thesis, it is a very important feature in open-domain conversational systems, as a poor TTS system quality can lead to poor quality of conversation. Additionally, more recent advances in the field can make a significant difference in the rapport-building

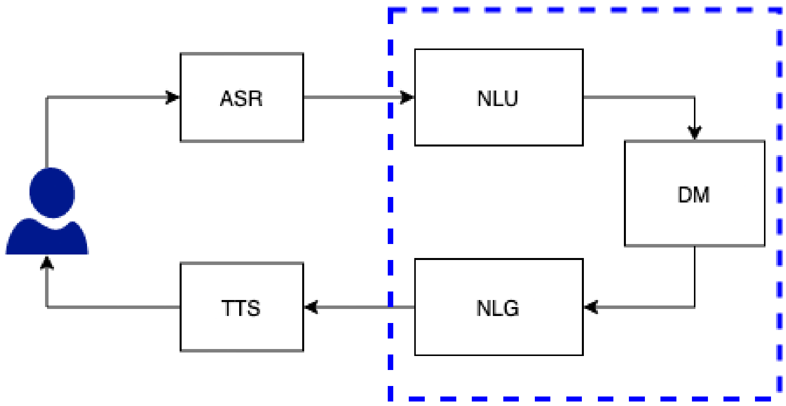


FIGURE 2.5: End-to-End architecture. The dotted rectangle denotes the modules that are substituted by the end-to-end neural model.

exercise that open-domain systems engage in, such as emotional-enabled TTS presented by Um et al. (2020), which could be well paired with the research in emotional conversational agents by e.g. Fraser, Papaioannou, and Lemon (2018).

2.1.2 End-to-End Architectures

E2E architectures are able to substitute the NLU, DM, and NLG components with a (usually single) DNN which has been trained on a large number of human conversations, drastically reducing the amount of manual work required to design and implement those individual components (Figure 2.5).

Transformer-based encoding can be utilised for more than classification tasks (such as user intent recognition). The ability of these models to encode the user's utterance with context, allows the design of E2E conversational systems without the need for meticulous and labour-expensive design of NLG templates and DM policies.

A few examples of such E2E systems include Meena (Adiwardana et al., 2020a), DialoGPT (Zhang et al., 2019b), BlenderBot (Roller et al., 2020), and BlenderBot 2.0 (Komeili et al., 2021).

Although such methods can yield really impressive results in terms of response naturalness and diversity (e.g. using the recent GPT-2 (Roller et al., 2020) or

GPT-3 (Brown et al., 2020a) models released by OpenAI), their responses often hallucinate facts and content (Dušek et al., 2020, Maynez et al., 2020) and often generate racist and unethical responses (Brown et al., 2020a, Gehman et al., 2020). This can be attributed to the *exposure bias* (Schmidt, 2019), where only ground-truth contexts are used during training but generated ones during testing, as well as noise in the data (Dušek et al., 2020) as training on web data contains bias as well. These limitations suggest that E2E systems are not yet suitable for general public deployment.

2.1.3 Ensemble Architecture

The aforementioned two architecture types cover most current dialogue systems. A third architecture category though, the ensemble architecture, allows the development of *multi-domain* dialogue systems, by combining rule-based, information-retrieval, or neural sub-modules. These sub-modules can be in the form of different 'bots', each specialising in a different domain (for example one bot could be specialised to provide retrieved news-related information, while another could provide social responses using generative models).

This architecture utilises the strength of all these different techniques and models when a dialogue system implemented using only one of them would falter. For example, generation-based models may provide fluent responses which are not always relevant, appropriate, or meaningful, and on the other hand retrieval-based models can provide very relevant responses which can be blunt and non-engaging (Chen et al., 2017).

Song et al. (2016) proposed a system where initially a retrieval module (using Lucene[4]) is used to retrieve the k-top most relevant responses to the user's utterance from a dataset of human conversations. Then a multi-Seq2Seq model, proposed by Zoph and Knight (2016), takes as input the relevant retrieved documents and the user's utterance in order to generate an additional new response.

[4] http://lucene.apache.org/

The generated and retrieved responses are then re-ranked based on a number of different features (such as entity similarity, length, and fluency) to retrieve the final response.

Zhuang et al. (2017) follows a similar approach using a retrieval mechanism to retrieve a coarse-grained shortlist of documents from a knowledge source (more information on how this mechanism works is described in the paper) followed by a number of different trained neural models re-ranking the list of candidate documents by their semantic similarity to the user's utterance. An additional multi-Seq2Seq model takes the user's utterance and a candidate document as input to generate a new response.

Tanaka et al. (2019) also combined neural generative with retrieval models. Their approach, however, includes an extension of Hierarchical Encoder-Decoder (HRED) (Sordoni et al., 2015) to encode the dialogue's context and the user's utterance in order to generate responses and a facts retrieval engine to retrieve relevant pre-encoded facts from a knowledge source. Then they use different techniques to promote more diversity and fact-related generated responses before they apply a ranking function to select the most appropriate one.

Concurrently with the development of Alana, most of the other teams in the 2017 and 2018 competitions were also developing similar conversational systems following a modular and ensemble architecture. An overview of the methods followed by those teams is outlined in Ram et al. (2018b).

Therefore, an ensemble architecture provides a better level of control than purely E2E regarding the system's output types, but requires an extensive amount of work to combine the different components together. Additionally, extending the system's functionalities (i.e. adding a new generator in the ensemble) needs careful design and retraining of the ranking function. However, from a design perspective, it still provides the highest level of extensibility and control in an open-domain setting, and it allows leveraging of different technologies that excel at different aspects of the dialogue (e.g. neural generated responses for chit-chat and information retrieval techniques for factual information and content delivery).

2.2 Evaluation of Open-Domain Dialogue Systems

2.2.1 Coherence and Response Generation metrics

To further investigate the **Research Question 2** (Section 1.2), a review of current evaluation metrics on coherence and engagement needs to be made.

Although in Human-Human Interaction (HHI) a number of the linguistic challenges discussed here and in Chapter 1 are handled almost subconsciously, artificial agents still lack a deeper understanding of the meaning of words, and therefore the ability to infer. Inference however is an integral part of any coherent dialogue. This makes the problem of coherence in conversational AI very challenging to approach.

The problem of evaluating responses produced by dialogue systems has been widely researched. Several methods have been introduced over the years to measure the quality and coherence of such responses, such as the centering theory (Grosz et al., 1995, Poesio et al., 2004) imposing restrictions on the distribution of discourse entities in coherent text (Cui et al., 2017). Barzilay and Lee (2004) and Fung and Ngai (2006) researched content approaches using HMMs which represent text as a sequence of topics and utilising topic shifts within a specific domain for assessing global coherence.

Elsner and Charniak (2008) in particular distinguish two levels of coherence: local and global. Local coherence represents how well connected neighbouring sentences are through lexical cohesion (Halliday and Hasan, 2014) or entity repetition (Grosz et al., 1995). Global coherence on the other hand, represents the relation between remote sentences (Kehler and Kehler, 2002).

A popular approach is the entity-grid model, in which the text is encoded into a set of lexical and syntactical properties (e.g., subject or object), followed by the employment of machine learning methods (e.g. SVM) for measuring coherence between these representations. Other entity-based characteristics are among the features investigated, like syntactic patterns (Louis and Nenkova, 2012), co-reference clues to ordering (Elsner and Charniak, 2008), and named-entity features (Elsner and Charniak, 2011). However, discovering and defining those characteristics is always an empirical process that necessitates a great deal of experience and subject knowledge.

Looking a bit more closely at the Entity-Grid model, it describes the structure of a dialogue using a grid that shows transitions in the syntactic roles of entities between adjacent sentences in the text. The rows in the grid indicate consecutive turns in the dialogue, while the columns represent each entity which can be a subject (S), direct object (O), or neither (X), with a symbol ($-$) to indicate that no entity appears in that turn t. The Entity-Grid model has been considered the state-of-the-art when it comes to calculating coherence in dialogue, with a number of researchers attempting to extend this model like Elsner and Charniak (2011), Guinaudeau and Strube (2013) and Filippova and Strube (2007). Cervone et al. (2018) also researched how the model can be extended with DA information to improve the model's performance especially when it comes to spoken dialogue (Figure 2.6).

	company	drugs	policy	convictions	clients
t1	S	X	-	-	-
t2	X	S	O	X	S
t3	-	-	-	-	-
t4	-	-	-	-	-
t5	-	X	-	-	-

A: *Entity Grid*

	company	drugs	policy	convictions	clients	*no_entities*
da1	qy	qy	-	-	-	-
da2	-	na	na	-	-	-
da3	-	-	-	-	-	sdê
da4	sd	sd	sd	sd	sd	-
da5	-	-	-	-	-	%
da6	-	-	-	-	-	qo
da7	-	-	-	-	-	nn
da8	-	sdê	-	-	-	-

B: *Modified Grid*

A. **t.1 da.qy** Well does [the company]$_S$ you work for test for[drugs]$_X$?
B. **t.2 da.na** Actually, they just recently started [a policy]$_O$ of testing [drugs]$_X$, which was kind of interesting,
B. **t.2 da.sdê** because when I went to work for them, uh, they didn't do that
B. **t.2 da.sd** but, uh, since then they've started a [drug]$_O$ testing [policy]$_O$, not because of their own, uh [convictions]$_X$, but because [the clients]$_S$ of [our company]$_X$ are requests that we do that.
A. **t.3 da.%** Huh.
B. **t.4 da.qo** How about you?
A. **t.5 da.nn** Uh, no
A. **t.5 da.sdê** we're not being tested for [drugs]$_X$ at all, uh

FIGURE 2.6: Entity grid example (A) vs. Cervone et al. (2018) suggested grid (B), as shown in Cervone et al. (2018). Entities in the sentences are annotated with their syntactic role: subject (S), object (O) or neither (X). The Dialogue Act tags are: qy (yes-no-question), na (affirmative-non-yes-answers), sde (Statement expanding y/n answer), sd (statement-non-opinion), % (uninterpretable), qo (open-question), nn (no-answers).

The current standard in evaluating Open-Domain Dialogue systems is to employ automated assessment metrics during model development and then use human judgement to evaluate the completed model. These are usually based on the notion of *word overlap*, finding common words between a given utterance and a candidate response (e.g. *overlap = 'the cat sat on the mat'* $\cap$ *'the cat sat next to the mat' = 4*). These automated measurements, however, have significant limitations and are not well correlated with human judgement (Liu et al., 2016, Novikova et al., 2017). Furthermore, human evaluation is too costly and time-consuming to be used during model development. Finally, a major challenge in evaluating open-domain conversation in particular is the *one-to-many problem*, where several plausible valid responses exist to a given utterance. That means that an evaluation metric could potentially penalise valid responses with very low word overlap. For example, as explained in Gupta et al. (2019), given a user utterance *"I would like to report a break-in"* and given different possible responses such as *"Was anything stolen"* or *"When was the break-in?"*, word overlap metrics would exclude the former although it is perfectly viable.

Some of these easily-applicable automatic metrics used in the assessment of the quality and coherence of open-domain conversations are:

F1 score. By taking the harmonic mean of the *precision* and *recall* (Equation 2.7 where tp is the true positive, fp the false positive, and fn the false negative examples), the F1 score calculates the word overlap between a generated sequence and the ground-truth (the gold standard response to the given utterance).

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{tp}{tp + \frac{1}{2}(fp + fn))} \tag{2.7}$$

BLEU, introduced in Papineni et al. (2002) is one of the most well-known word overlap metrics in NLG based on n-gram precision calculation between a generated sequence and a reference/context. However, BLEU penalises shorter sequences given that precision (Equation 2.8) favours shorter sequences since a longer candidate is more likely to contain a larger fraction of the reference than a shorter candidate.

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \tag{2.8}$$

Furthermore, according to Liu et al. (2016), Lowe et al. (2017) and Gupta et al. (2019), BLEU also does not correlate well with human judgement.

METEOR (Denkowski and Lavie, 2014) shows some improvements over the BLEU score, which does not favour specific sentences over others (e.g. longer over shorter sentences), by using a weighted F-score based on mapping unigrams as well as a penalising incorrect word order. METEOR also includes additional features like stemming and synonymy matching compared to other metrics, and was also found to correlate better with human judgement.

ROUGE[-N/L/W/S] (Lin, 2004) is used mainly for summary evaluation and is based on word overlap. There are various different sub-types of this metric depending on this overlap. **ROUGE-L** for example calculates the longest common sub-sequence between a generated output and a reference.

Greedy Matching (Rus and Lintean, 2012) is based on semantic similarity approaches, where it greedily matches each generated output token's embeddings to a reference word using their cosine similarity. Finally the generated sequence's score is the average of all individual token words.

Embedding Average (Wieting et al., 2015) uses a similar approach to Greedy Matching, but instead of calculating the embedding similarity of a given text word-by-word, calculates the cosine similarity between the *average* word embeddings of the entire generated sequence and the *average* word embeddings of a reference (ground-truth).

Vector Extrema (Forgues et al., 2014) is an alternative method for calculating sentence-level embeddings, particularly useful on dialogue tasks as shown by Gupta et al. (2019) and Liu et al. (2016), by calculating the similarity between the *maximum* value of each dimension of the word embedding of the generated and the reference sequences.

BERTScore (Zhang et al., 2019a) is one of the more recent metrics, increased in complexity, that shows improvement in correlation with human judgement. It is based on similar approaches to the above embedding-based metrics, as in it calculates the cosine similarity between each token of the generated sequence $\hat{x}$ against each token of the reference sequence x. However, it extends those approaches by employing contextual embeddings (Devlin et al., 2018) instead of exact matching using simple word embeddings (Mikolov et al., 2013c) (Figure 2.7). BERTScore scores the recall of the generated sequence. However the full metric is comprised of all three relevant scores (*precision* P_{BERT} (Equation 2.10), *recall* R_{BERT} (Equation 2.9), and F_{BERT} (Equation 2.11)).

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^\top \hat{x}_j \tag{2.9}$$

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_i \in \hat{x}} \max_{x_j \in x} x_i^\top \hat{x}_j \tag{2.10}$$

where x_i and $\hat{x_j}$ denote a token in the reference sequence and a token in the candidate sequence respectively.

$$F_{BERT} = 2\frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}} \tag{2.11}$$

Additionally to the above, there have been several promising metrics recently that require no to very few human annotations that correlate reasonably well with human judgement:

Dziri et al. (2019) recommend assessing open-domain dialogue systems by looking at how consistent the generated response is to a given context. They refer to the automated dialogue evaluation as an **entailment** problem. In order to turn

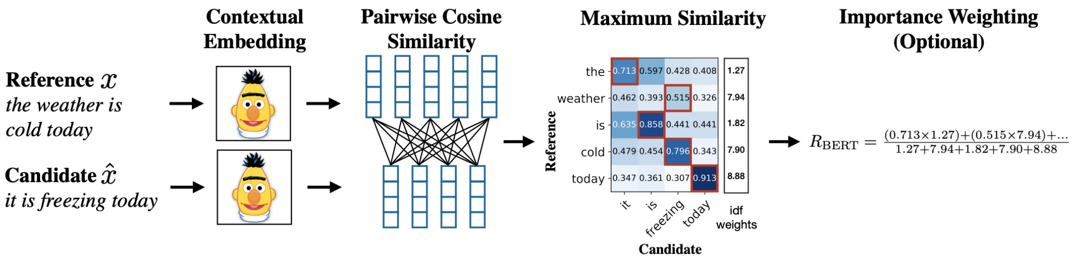


FIGURE 2.7: Computation of the recall metric R_{BERT} as illustrated in Zhang et al. (2019a). Given the reference utterance x and candidate $\hat{x}$, contextual embeddings are used representing the utterance tokens, and cosine similarity is employed to compute possible matches.

the automatic evaluation into an Natural Language Inference task, the authors treat the generated response r as a hypothesis and the conversation history H as a premise and recognise whether the response is derived from the history of the conversation (whether it *entails*, *contradicts*, or is *neutral* to the premise). They use synthesised inference data from conversational corpora to train state-of-the-art inference models.

Mehri and Eskenazi (2020) introduced an UnSupervised and Reference-free (**USR**) metric to tackle problems with evaluating open-domain dialogues, especially given the aforementioned one-to-many nature (multiple valid responses to a given user utterance) of such dialogues (Zhao et al., 2017). The metric consists of several sub-metrics which combined evaluate the overall quality. According to the authors, the strength of this approach lies in the fact that the metric does not rely on some provided *ground-truth*, rather that an unsupervised model is trained to measure specific properties in the dialogue. The properties measured are *understandable, interesting, natural, maintains context.*

Inspired by the alignment measuring in Natural Language Inference as described in Dziri et al. (2019), Sinha et al. (2020) introduced an unsupervised and reference-free metric called **MaUde** (**M**etric for **a**utomatic **U**nreferenced **d**ialog **e**valuation), which instead of measuring *entailment* or *contradiction*, uses a trained classifier to evaluate the quality of a generated response given a conversation history as context. In other words, given a context c the model used in this metric is trained to distinguish between a correct response r, $(score(c, r) \rightarrow 1)$, and a false response $\hat{r}$ $(score(c, \hat{r}) \rightarrow 0)$.

More recent work in evaluating open-domain conversations, includes the **Sensibleness and Specificity Average (SSA)** metric, used in the chatbot Meena (Adiwardana et al., 2020b), which uses human judgement to evaluate sensibility and specificity (how specific the responses are given a conversation context) of the responses.

Similarly, Thoppilan et al. (2022) are using human raters to evaluate *quality (measuring sensibleness, specificity, and interestingness of the responses), safety, and groundness* on a given context.

2.2.2 Engagement metrics

Engagement is one of the most critical metrics when evaluating open-domain conversational AI dialogue systems (Ghazarian et al., 2020, Pamungkas, 2019, See et al., 2019, Yu et al., 2016b). Most research uses automatic features such as number of turns or time duration of the conversation to assess how engaging a dialogue is. During the Amazon Alexa Prize Competition (which will be described in the next chapter) evaluation of the user's engagement was based on the number of turns and total duration of conversation (while the quality of the conversation was measured by an overall subjective user rating at the end of the conversation) (Venkatesh et al., 2018a). This engagement metric however assumes that lengthier dialogues correlated positively with engagement, when in reality a longer dialogue could occur as a result, for example, of misunderstandings or repeats due to an inability of the system to understand the user's intents (Ghazarian et al., 2020).

Ghazarian et al. (2020) proposed an automatic metric to predict engagement on an utterance level by training a transformer-based classifier taking a user's utterance and the system's response and predicting whether the response is engaging or not. Yu et al. (2016a) used a Wizard-of-Oz approach to train an utterance-level engagement classifier, which was then used by DM policies to optimise subsequent dialogue turns if the previous one showed a decrease in user engagement.

2.3 Task-based metrics

So far the metrics described in this chapter were evaluating the quality of system generated responses primarily in terms of their coherence to the dialogue's context which in certain cases could be a good estimator for purely social dialogue. However, as will be described in Chapter 7, open-domain capabilities can be weaved into task-oriented systems to increase user satisfaction. In that case additional metrics evaluating the task-based dialogue need to be applied.

A paradigm framework for task-based spoken dialogue system evaluation, PARADISE (Walker et al., 1997), is able to evaluate the overall user satisfaction using a weighted function of task success measures and dialogue cost (which can be split into dialogue efficiency - e.g. number of utterances, and dialogue quality - e.g. system response delay) intended to identify features affecting the user's perception of the system (Hajdinjak and Mihelič, 2006). The PARADISE framework includes a user-satisfaction survey to capture subjective user satisfaction and perception on the performance of the system. The survey includes questions aimed to evaluate the system's *ASR and TTS performance*, the *task ease*, *user expertise*, system's *expected behaviour*, *future use*, etc. These subjective measures alongside the objective measures (task success and dialogue cost) are passed to a linear regression model to predict the overall user satisfaction. The Amazon Alexa Prize Competition (Venkatesh et al., 2018b) utilised the *future use* survey questions of the PARADISE framework to evaluate the overall rating of a conversation. At the end of each dialogue, the user was asked *"Would you want to talk to this social bot again in the future?"* and was prompted to answer this using a Likert scale between 1 (definitely no) and 5 (definitely yes).

The PARADISE framework, however, evaluates the user satisfaction on the dialogue as a whole. Sometimes it is useful for the evaluation to be on a turn level allowing a more fine-grained optimisation of systems and evaluating different aspects of the dialogue as it progresses. This can be done either automatically by annotating the dialogue on an exchange level either by users directly (Engelbrecht et al., 2009) or by domain experts (Higashinaka et al., 2010, Schmitt and Ultes, 2015). An alternative option for automatic evaluation on a turn-by-turn basis was proposed by Shalyminov et al. (2018) using sentiment analysis to extract the user's satisfaction of the system's response at any given time. Additional features such as dialogue length were used as part of the training of a neural ranker as a selection strategy to an ensemble conversational system. Further description of this ranker will be presented in Section 3.4.

Hara et al. (2010) suggested an alternative method to predict task-success rate by utilising ratings on a dialogue-level (like PARADISE) but as a target output of a trained model and n-consecutive DAs as input.

More recently, Følstad and Taylor (2021) suggested a framework for qualitative analysis of different dialogue systems similar to the key concepts presented in the PARADISE framework, focusing on response relevance and understandability as well as dialogue efficiency and outcome. The framework however was evaluated on a small number of customer service chatbots, which requires further investigation on how it would perform in open-domain dialogue systems.

2.4 Conclusion

In this chapter different conversational AI system architectures were explored, primarily focusing on modular and end-to-end architectures. Additional combination and hybrid architectures were not included in this review (such as E2E task-oriented systems optimised with RL by Schmidt (2019)). The different components' functionality in a modular architecture were described, focusing on the differences, advantages, and limitations when applied to task-oriented and open-domain conversation alike. Further distinction was made in terms of the implementation of these modules, between rule-based and neural approaches discussing the advantages and disadvantages each one brings. Furthermore, statistical methods using RL were discussed, enabling the optimisation of the policies used by the DM dictating the actions of the system on each subsequent turn.

Although the use of POMDPs allows the optimisation of more complex policies where uncertainty is an issue, in open-domain conversations the action and state space is practically infinite. Thus, statistical approaches were not considered as an overall DM implementation while designing Alana as will be described in the following chapters. However, as will be described in Section 5.3, a statistical approach was used to optimise smaller, more well-defined policies.

E2E, an alternative to the modular architecture was also described, substituting most of the aforementioned components with a DNN. With this architecture most of the designer's decisions (and control) are handed over to the neural network, learning the intricacies of the human language by training on real conversational data. This method provides natural results, however, it was also shown that the results can be quite unpredictable or not grounded on the user's queries, features required to hold a coherent and engaging open-domain conversation between the conversational system and the users.

Since an open-domain conversation should intrinsically be *multi-turn*, the responses the system provides need to be coherent. A variety of different evaluation metrics for coherence and quality of response generation was described, including widely used metrics such as BLEU and Entailment.

In the next chapter, Alana, a modular-ensemble system developed during the Amazon Alexa Prize competition in 2017 is described, showing how the aforementioned techniques and technologies can be combined to tackle the variety of challenges present in open-domain conversation. As was discussed in this chapter, a modular approach still is the most controllable and practical way to develop an open-domain Spoken Dialogue System and still allows individual components to be developed in a data-driven way. Thus, multiple specialist bots can be developed in such an ensemble to provide a wider coverage of functionality, topics, and response variance in an open-domain conversation.

Chapter 3

The Alana v1.0 conversational Framework

In this chapter, an ensemble and modular dialogue system, called Alana, is presented. This version of the system presented here is the first attempt to tackle the challenges outlined in Section 1.1 and answer **Research Questions 1** and **4** (see Section 1.2). Initially, the Amazon Alexa Prize competition aims and objectives are described, which the Alana system was optimised for. Later in this thesis (Chapter 5) an updated version of the system described below is presented.

3.1 The Amazon Alexa Prize

On September 26, 2016, Amazon announced a worldwide university competition called Alexa Prize[1], aiming to advance research in Conversational AI (Ram et al., 2018a). Through this competition, participating universities were able to build *open domain* conversational agents, called "social bots" to conduct research and test hypotheses. Those social bots were deployed to the Amazon US marketplace, where anyone with an Alexa enabled device was able to interact with them by saying "Alexa, let's chat".

[1]https://developer.amazon.com/alexaprize

The challenge was the same across all teams, which was to build the social bot in such a way as to be able to engage in an open domain conversation with their users over various popular topics, such as Sports, Politics, Entertainment, Fashion and Technology for prolonged periods of time. The interaction however should be engaging and entertaining to the users. The grand winner would be the social bot that would remain engaging and coherent in two conversations of over 20 minutes. Once the user invokes the Alexa Prize skill (through "Alexa, let's chat") he would be redirected to one of the participating social bots randomly. The teams were instructed not to reveal the University's/team's identity throughout the conversation at any point to maintain anonymity and fairness in the competition. This lead to some of the decisions made in the design of the system as are described below. At the end of each conversation, the users were allowed to leave a **single rating** on the quality of the conversation using a 5-point Likert scale as well as feedback to the socialbot's competing team.

Each team was given access to a variety of AWS resources including GPU-enabled virtual machines and access to previous Amazon user statistics (e.g. most common topics/phrases used during user interactions with Alexa). Each social bot used Amazon's provided ASR, which was developed for the purposes of the competition using a custom Language Model (LM), specifically designed and fine-tune to handle open-ended conversations (Ram et al., 2018a). The social bots' design, architecture, functionality, and deployment was entirely up to each individual team.

Producing an engaging, coherent, and entertaining open domain conversation is quite challenging, with a number of challenges outlined in section 1.1 that need to be addressed. A few of the most challenging tasks addressed in this chapter are:

- **Maintaining the context between the turns of each interaction**. Although in recent years a lot research has been made in generating and evaluating contextually correct responses across a small number of turns, as described in section 2.2, longer conversations tend to require a much more extensive context memory. Generative systems such as GPT-2 (Radford et al., 2019) are trained to produce extremely natural responses, however they fail to utilise context of more than 1 turn (Li et al., 2017).

- **Provide coherent responses**. Another challenge is that in longer open domain conversations, the interlocutors tend to talk about different topics, often linked with each other in a meaningful way (e.g. when discussing about a recent movie the discussion could easily shift to related news about one of the actors). So the system should not only reply in a coherent way with regards to the current topic, it also needs to be able to perform topic switching in a natural way to keep the conversation interesting and engaging.

- **Be engaging**. Given the context of the competition, to be able to hold a conversation for prolonged periods of time, it is paramount that the users want to continue interacting with the system. Duration alone is a poor metric for such interactions, as even a task-oriented system can prolong a conversation by simply repeating itself or not attending to possible speech recognition errors (Ghazarian et al., 2020). This can easily lead to the user's frustration and boredom, possibly leading to the user ending the conversation. Especially in an open domain environment, where the user has the freedom to discuss anything they want, the responses provided by the system need to be entertaining, relevant, correct, and attentive to the user's interests. Furthermore, if the conversation is not engaging enough, the users quite often do not know how to response to a system's response in order to continue the conversation. In these cases, the system should be able to help drive the conversation forward.

- **Be knowledgeable**. When building an open domain conversational system one must make sure that the responses provided by the system are not only coherent and engaging, but also correct and relevant. That requires an up-to-date access to real world data sources (such as news, wiki articles) accessible in a way that the system will be able to utilise during the conversation.

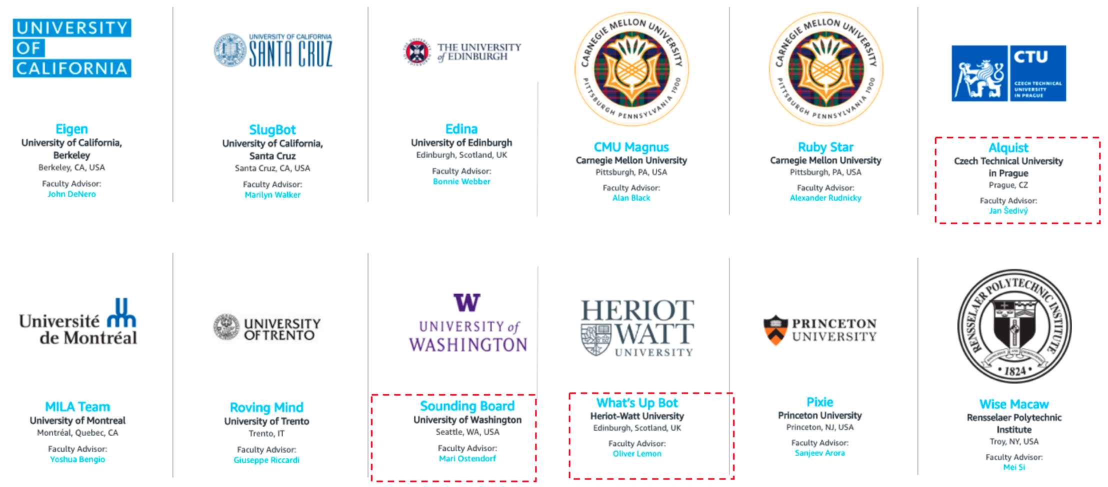


FIGURE 3.1: Alexa Prize 2017 competing teams. 100 entrants, 12 semi-finalists, 3 finalists (denoted by dotted box). Entry to finals based on user score over 6 week semi-final.

3.2 Overall System Architecture

Our team (called *What's Up Bot* for the purposes of the 2017 competition) consisted of 6 PhD students (Ioannis Papaioannou, Amanda Cercas Curry, Jose L. Part, Igor Shalyminov, Xinnuo Xu, and Yanchao Yu) and 3 faculty advisors (Oliver Lemon, Verena Rieser, and Ondřej Dušek). The work presented in this chapter is the outcome of the joined efforts of the team during the Amazon Alexa Prize 2017. As a team leader my work and contribution, primarily focused on the system architectural design, as well as the design, implementation, and evaluation of CoherenceBot, IntroBot, RapportBot, and user modelling capabilities, as described in this chapter. The team's competing system (henceforth called *Alana*) reached the finalists stage (among 100 entrants - Figure 3.1).

The team's approach follows the work of Yu et al. (2016b), using a collection (ensemble) of different bots, each generating a different type of candidate response (or multiple responses) and a ranker to select the most appropriate response from the pool on each turn in the conversation.

The overarching vision we had for our system, called *Alana*, was to imitate the type of conversation someone would they have with someone who just met (e.g. at a pub), where the participants would casually talk about the news, exchange fun facts, talk about their interests, etc. One common aspect in these sort of scenarios is that usually each participant is trying to discover the other participant's interests, ultimately resulting in a subset of mutually interesting topics that create a pleasant conversation.

Following this assumption, within the Alana system, we focused on keeping the interaction humorous and engaging, while trying to stick to topics that the user was interested in (as explained in more detailed in section 5.2.4). The system was also built to not only be able to converse about the topics that the competition was requesting (Sports, Politics, Entertainment, Fashion, and Technology), but engage in completely open-domain conversation with its users. Another important feature to provide more natural conversations with the system is to allow the user to freely and abruptly change topics or otherwise change the direction of a given conversation at any time. For example, if the current topic of discussion is about movies and the system just asked the user a question (e.g. *"Do you prefer sci-fi or comedies?"*) the conversation should not break if the user escaped that conversation with something like *"actually is there any news about coronavirus?"*

3.3 System's Pipeline

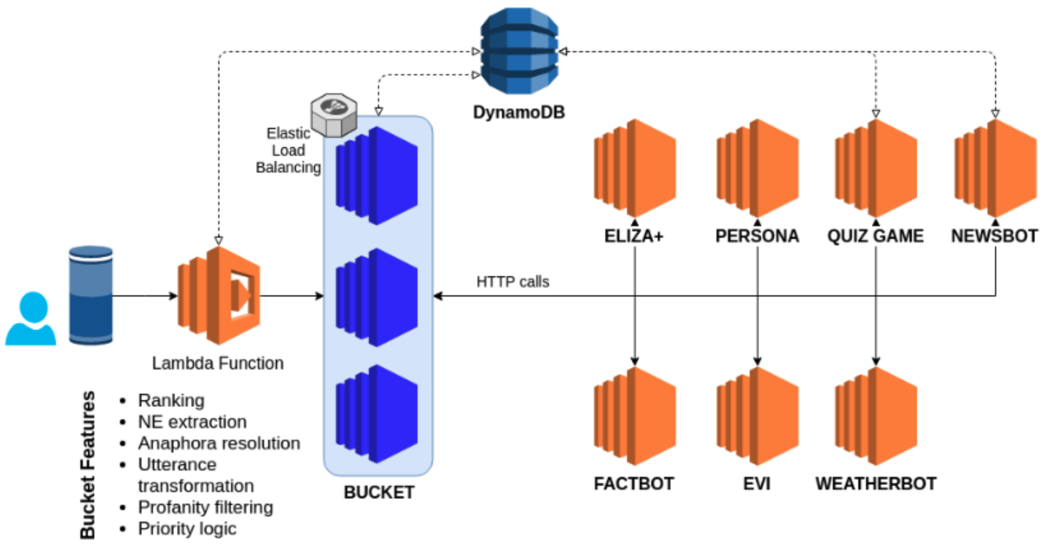


FIGURE 3.2: Alana's hybrid hierarchical architecture with ranking

3.3.1 Lambda Function

The system architecture is shown in Fig. 3.2. Since the social bots were accessed through Amazon Alexa enabled devices (such as Amazon Echo), our system was receiving the event containing the user's transcribed utterance through an AWS lambda function. Through the lambda function, we were also able to receive a couple of Amazon pre-trained intents, such as the *StopIntent* (which captures any form of stop-related phrases). Other useful metadata such as the timestamp and the tokenised ASR confidence scores (the ASR scores of each individual word in the sentence), were also included in the event. This data is stored in a DynamoDB database and then forwarded to a load-balanced Amazon EC2 instance, henceforth called the *Bucket*, which runs the main logic pipeline of the system.

3.3.2 The Bucket

The Bucket is the backbone of the entire system featuring a number of different operations from the moment it receives the information from the Lambda function until the final result is output back to the user.

Here, the user's utterance undergoes several NLU pre-processing steps such as standard textual syntax for representing patterns for matching text, called regular expressions or regex. Regular expressions provided a simple, rapid, yet powerful method of generalising the users' language on selected intents (e.g. *"let's talk about* $\{X\}$*"* or *"My name is* $\{Y\}$*"* where X and Y would be the parameter of the intent) especially at the beginning of the competition where real user data was not yet available.

For example, for the intent *tell_me_about* where the system would extract the topic or entity the user wishes to discuss, several regular expressions such as in Listing 3.1 would be written.

```
tell_me_about:
    "- can (?:we|you) (?:talk|chat) (?:about )?((?!about).+)"
```

```
"- can (?:we|you) have a conversation about (.+)"
"- can you tell(?: me)?(?: something)? about (.+)"
...
```

LISTING 3.1: Example regular expression for the *tell_me_about* user intent. (.+) allows extraction (also known as grouping in regex) of any substring placed there that can be used as the parameter of the intent - in this case the topic or entity to talk about.

The ASR's confidence score received by the Lambda function was also checked in the Bucket. If the confidence falls below an empirically set threshold (set to 0.25[2]), the system prompts the user to repeat themselves (e.g. *"I am sorry. I think I heard you said [user's utterance as understood by the ASR]. Could you repeat that please?"*).

The utterance is then further processed for ellipsis (Johnson, 2001) and co-reference phenomena.

As explained in Chapter 2, ellipsis resolution puts a fragmented sentence into the right context based on the history of the conversation. For example, the user's utterance on a specific turn might be *"Yes"* as a response to the system's last question *"Do you like ice-cream?"*. *"Yes"* in itself without the context does not convey any useful meaning, in which case this process will try to restructure the utterance to include the context. Thus, the transformed utterance will turn into *"Yes I do like ice-cream"* which makes it easier for downstream tasks (i.e. the bots to be called) to be completed.

Co-reference resolution is the task of matching all expressions in a given text referring to the same entity. For example in the sentence *"My friend **Mary** said **she** can't make it tonight"*, *she* refers to *Mary*. This is a crucial part of the preprocessing in multi-turn conversations where the entity in focus changes frequently.

[2]The threshold was set after experimenting with various values daily. The team concluded that this value was sufficient enough to perform well without excluding a substantial amount of utterances.

Further processing of the utterance follows allowing (a) transformation of indirect questions to direct questions (e.g. *"I don't know who Han Solo is"* → *"Who is Han Solo"*) as well as certain types of replies (e.g. *SYS: "What is your favourite movie?", USR: "The Man who Cried. How about you?"* → *"What is your favourite movie?"*). All of these transformations are implemented using mostly regular expressions. Finally, Noun Phrases (NP)[3] and NE[4] are extracted from the utterance using Stanford's CoreNLP library. Both NP and NE are saved in the database in order to be available during the following turns (for co-reference and ellipsis resolution).

This information is then forwarded along with the preprocessed utterance and 3 pair of turns (a user and a system response) of dialogue history to a collection of bots, each running on a different EC2 instance. The communication between each both and the Bucket is done via HTTP GET calls[5]. All the extracted information from the user's utterance is encoded in the call's URL string (e.g. *https://url/to/bucket?q=foo%bar&sid=test_session&u=test_user*).

The full list of parameters available to the bots is:

```
'q': user utterance,
'nnq': flag denoting utterance does not contain any named entities,
'p': part-of-speech filtered utterance,
'e': list of named entities,
'sid': session id,
'lb': responded bot name,
'u': user name,
't': current topic,
'pr': list of user preferences,
'n': turn number,
'i': user intent
```

[3]https://stanfordnlp.github.io/CoreNLP/pos.html

[4]https://stanfordnlp.github.io/CoreNLP/ner.html

[5]This method was decided in order to make use of the *flask* Python library that the team was more experienced with

Each bot in the architecture acts independently of each other and is able to use what information received from the Bucket as they see fit. They then produce and return one or more candidate responses based on their internal functionality and purpose.

Once all of the candidate responses by the bots have been gathered in the Bucket, the DM of Alana (which in this version of the system was primarily a ranker) selects the most appropriate response from the pool based on a number of different metrics as described in Section 3.4.

3.3.3 The Bot Ensemble

Alana's collection of bots in this version can be divided into three categories:

- **Rule-based** bots, which are used to respond to the user in a controlled and consistent way.
- **Information-retrieval** bots, where they have access to pre-indexed data from a variety of data sources, such as WashingtonPost, movie databases, wikipedia, etc. Information retrieved from these sources is then wrapped in hand-engineered templates to be delivered to the user in a more conversational way. These bots provide the driving force that allows the conversation to flow from topic to topic and provide the user with the freedom to discuss any of them.
- **Miscellaneous** bots, that perform very specific tasks, such as playing a game with the user, or delivering fun facts. A couple of API-based bots are also included here, where the information is retrieved live from an external API source (such as weather information).

The final ensemble included the following bots. Three additional bots (IntroBot, RapportBot, and CoherenceBot) are described in more detailed in the following section.

3.3.3.1 PersonaBot

A rule-based bot written in AIML[6]. This bot maintains personality related responses consistent throughout the conversation, such as the system's movie or music preferences. As opposed to a similar system described in Li et al. (2016a), PersonaBot is purely rule-based and not generative, in order to allow control over replies to any situation where a specific response is expected, such as suicide-oriented user utterances where in this case the system is expected to provide contact information to the Samaritan hotline[7], as per the competition rules.

Example responses

S: Hi, [...] What would you like to talk about?
U: movies
S: *(PersonaBot)* Great! What is your favourite movie?
U: The Lord of the Rings.

3.3.3.2 ElizaBot

An extension of the ELIZA-style (Weizenbaum, 1966) chatbot called *Rosie.*[8], customised for the competition from the original mobile device-oriented version.

Example responses:

U: You know what? I'm really excited about dogs!
S: *(ElizaBot)* I'm excited about dogs too!

3.3.3.3 NewsBot

An information retrieval bot based on the open-source framework Lucene[9] that is able to talk about the news. The bot indexes and searches a variety of news

[6] http://www.alicebot.org/aiml.html
[7] https://www.samaritans.org/
[8] https://github.com/pandorabots/rosie
[9] https://lucene.apache.org

article both from Washington Post (api access provided by Amazon), as well as selected news sources provided via NewsAPI[10]. The articles are then automatically summarised using the *Sumy*[11] library. This step is important as the system's deploying platform is voice interactive, and longer text could be harder to understand through an auditory channel. The selection of the relevant summarised articles is done using the BM25[12] algorithm to score news relevance on 1, 2, and 3-grams over the query and dialogue context. Empirically designed weights on NP and NE extracted from the user's utterance were also able to boost the selection of the right article. Finally, the 10 most relevant articles (scored by Lucene) are further ranked based on the recency of the articles. That enabled the system to retrieve the newest articles or answer user queries like *"What is the latest news?"*.

Example responses

U: What's happening with **Bob Dylan**?

S: *(NewsBot)* I heard this on the radio – Was Dylan too freewheelin' in borrowing for Nobel lecture? The whiff of plagiarism is blowin' in the wind for Bob Dylan. Want to know more?

U: sure

S: *(NewsBot)* It seems that you are interested in this topic. I'm so glad. Here's more. Here's Dylan: Some men who receive injuries are led to God, others are led to bitterness The cases Pitzer found are not blatant or explicit – there are no verbatim sentences, only identical phrases and similar phrasing. What are your opinions on Bob Dylan?

3.3.3.4 Factbot

A rule-based bot delivering fun facts, stories, and jokes from a collection of various different online resources. The responses are delivered upon explicit intent from the user (e.g. *"Can you tell me a joke?"* or even containing a particular NE (e.g. *"Tell me a fact about X"*). This bot was also the safety net and deflection

[10] https://newsapi.org
[11] https://pypi.python.org/pypi/sumy
[12] http://ir.dcs.gla.ac.uk/wiki/BM25

strategy of the system if none of the other bots in the ensemble were able to provide a suitable response to the user's query.

> Example responses
>
> U: Can you tell any jokes?
>
> S: *(FactBot)* How did the hipster burn his mouth? ... He ate the pizza before it was cool.

3.3.3.5 Quiz Game

A rule-based system developed using VoiceXML[13]. This bot aims at entertaining the user to maximise engagement with the system. The user is presented with topic-specific questions (e.g. 80s music, science, history, sport, geography) and they have to guess the right answer from the provided options (e.g. *"Which 80s Clash song, when re-released in 1991, went straight to number one in the UK? Should I Stay or Should I Go? London Calling? Straight to Hell? Or Rock the Casbah"?)*).

> Example responses
>
> U: Let's play a game
>
> S: *(QuizBot)* Great, let's play. please remember, whenever you want to leave the game, just say end game. Now, we have several topics, 80s music, science, history, sport, geography, which one would you like?
>
> U: history
>
> S: *(QuizBot)* Got it, you selected history, next question. In what industry did John Davidson Rockefeller get rich? Motor, Music, or Oil?
>
> U: oil
>
> S: *(QuizBot)* You are right! Next question. What was the first name of the US President known by the nickname JFK?, Jerry, John, or Joe?

[13]https://en.wikipedia.org/wiki/VoiceXML

3.3.3.6 EviBot

A bot retrieving factual information (e.g. *"Who was the 4th president of the United States?"*) based on the user's query, powered by the EVI question answering engine API[14], provided by Amazon.

Example responses

U: [***Talking about Bob Dylan in the previous turns***] When was **he** born?

S: *(EviBot + anaphora resolution)* Bob Dylan was born on Saturday May 24th 1941.

3.3.3.7 WeatherBot

A simple rule-based bot that provides the user with weather-related information, if asked for, querying the *OpenWeatherMap API* (https://openweathermap.org/) on the fly. Given that no personal information was collected from the users (apart from any given name), Amazon's Seattle Headquarters was selected as a default location. If the user enquired about the weather to a specific location that information would be given instead.

Example responses

U: What's the weather today?

S: *(WeatherBot)* Right now in Seattle, it is 74°F.

3.3.3.8 WikiBot

An information retrieval bot with access to Wikipedia data[15], based on Apache's Lucene[16] index and search framework. An index was created using the headline and the first sentence of each paragraph of each entry in Wikipedia. This bot

[14] https://www.evi.com/
[15] Data retrieved then indexed from https://dumps.wikimedia.org
[16] https://lucene.apache.org

was triggered when both (a) the *tell_me_about* user intent (see Section 3.3.2) was recognised, and (b) NP or NE were present in the user's utterance. The WikiBot retrieved the top scored document in the index containing these entities.

Example responses

S: *(CoherenceBot)* Speaking of video games. So what is a video game that you like? I love Minecraft.

U: overwatch

S: *(WikiBot)* Ok, I think you said overwatch, if I'm not mistaken, Overwatch is a team-based online multiplayer first-person shooter video game developed and published by Blizzard Entertainment. It was released in May 2016 for Microsoft Windows, PlayStation 4, and Xbox One. Let's chat more about Overwatch.

U: okay

S: *(WikiBot (multiturn))* Overwatch is a first-person shooter that features squad-based combat with two opposing teams of six players each. Players choose one of several heros [...]

3.3.3.9 Intro Bot

Coherence and engagement in this version of Alana were maintained using 3 different bots as part of the bot ensemble: *Intro bot*, *Coherence bot*, and *Rapport bot*.

The IntroBot's functionality is very specific, as it only replies during the first couple of turns of each conversation. It performs a simple slot-filling task with the purpose to greet the user and ask for their name, which could then be used throughout the conversation. The IntroBot also analysed the user's utterance for a particular sentiment (we used NLTK Vader[17] sentiment analyser for this task). This information was used to provide a more natural response when the user was asked "How are you doing" related questions (See examples below).

[17] https://www.nltk.org/api/nltk.sentiment.html

During the next turn, the user would be prompted to give their name, if they so chose. To keep the interaction enjoyable and make the user feel they are having a more personal conversation with the system, special attention was paid to allow the user to provide their preferred (non-profane) name to be addressed (e.g. *"My name is John but my friends call me Darth"* → *user_name: Darth*)

Example responses with happy and compliant user

S: *(IntroBot)* Hi. How's it going with you?
U: I am feeling great! [***Positive sentiment detected***]
S: *(IntroBot)* Awesome! I'm glad you are doing well today! If you like, you can tell me your name?
U: Call me Tim [***User gave their name***]
S: *(IntroBot)* Pleasure talking with you Tim. Let's get to know one another. *+ (response from RapportBot)*

Example responses with unhappy and non-compliant user

S: *(IntroBot)* Hi. How are you feeling today?
U: I am feeling down today. [***Negative sentiment detected***]
S: *(IntroBot)* Oh dear. I'm sorry to hear that. Perhaps I can make your day better! So. Could you tell me your name?
U: I don't want to tell you my name. [***User did not give their name***]
S: *(IntroBot)* Okay. Let's get chatting then!. *+ (response from RapportBot)*

Once the conversation would reach that point, the system would call the *RapportBot* directly and append that bot's response to the IntroBot's answer.

3.3.3.10 Rapport Bot

This bot also performs a slot-filling task with a twofold functionality. It was (a) explicitly trying to find some of the user's preferences by asking direct questions, information that was then able to be used by any of the bots in the ensemble in subsequent turns, and (b) trying to prime the conversation on subjects that could be of interest to the user and easier for the system to handle. This bot was triggered either by the *IntroBot* directly, as explained in section 3.3.3.9, or by the user's intent (eg. *"Let's get to know one another"*). The bot is based on a pattern-template (using regular expressions) architecture, where each question is associated with a set of predetermined user answers that are able to trigger the next response of the bot. Those user responses are captured using regular expressions as shown in Listing 3.2.

```
- question: "Are you more interested in reading books or watching
 movies?"
options:
- preference: books
response: "Oh! A book worm! How nice. I like reading books on the
 cloud. Let's see, what else... "
pattern: "\\b(read(ing)?|books?)\\b"
- preference: movies
response: "A movie fan, huh? There are some really nice movies out
 lately! Personally, I enjoy audiobooks for obvious reasons... But
 let's move on. "
pattern: "\\b(movies?|films?)\\b"
- preference: both
response: "Awesome! Personally, I enjoy audiobooks for obvious
 reasons... But let's move on. "
pattern: "\\b(either|both|do(n't| not) know)\\b"
- preference: none
response: "Alright! Let's move on then. "
pattern: "(none|neither|any|(do(n't| not)
 (like|prefer))|hate|dislike|tell me about something else)"
```

LISTING 3.2: In this example *question*, the user is expected to respond with either of the options (*movies* or *books*), both, or neither of them (explicitly)

It is worth reiterating that none of these prompts were "locking" or compelling the user to answer, and the user was completely free to "escape" and change topic at any time. Further examples of those patterns and templates are shown in Appending A.1

The user's preferences are then stored in the DynamoDB by the RapportBot so that the information can be used by the rest of the ensemble in subsequent turns.

3.3.3.11 Coherence Bot

This bot keeps track of the current topic discussed (from a set of predetermined topics that it can handle - as shown in table 3.1), trying to keep the user engaged on the current topic, or it suggests a new one (primarily based on the user's preferences). Additionally, it helps drive the conversation forward, thus retaining the flow of conversation, using one of the pre-scripted appropriate responses (which we call conversation *drivers*). This is particularly helpful in situations where (a) the user doesn't know what else to say (which is a frequent phenomenon when users who are not familiar with speech technologies are prompted to discuss about anything they like) and (b) handling Out-Of-Domain (OOD) utterances if the user's input doesn't contain any useful information for the rest of the bots in the ensemble to provide a suitable answer. Those drivers are selected either based on the current topic or the user's preferences that were captured by the *RapportBot*. For example, if the current topic is *books*, an appropriate driver would be *"What is a book that you like a lot?"*. However, it quickly became apparent during the competition by looking through the system logs (see Chapter 4) of the interactions with the users, that usually they were not able to maintain the conversation on a given topic, especially when directly asked a personal question. To mitigate this and keep the conversation flowing naturally, CoherenceBot's drivers include the bot's "personal" (handcrafted) opinion on that question to help the user with an example. In the book example above, the complete driver would be in this case *"What is a book that you like a lot? I love, do androids dream of electric sheep."*. This allowed a controllable and safe recovery of the conversation when no specific intent was recognised that would carry the conversation forward. A similar approach was followed by Konrád et al. (2021) in the 2021 Alexa Prize competition using a neural-trained bot instead in order to handle OOD utterances.

movies	celebrities	architecture	cars
book	animals	artificial intelligence	gardening
technology	food	pets	space
music	fashion	relationships	sci-fi
philosophy	politics	tv shows	fantasy
science	video games	mars	sewing
sports	art	veneus	

TABLE 3.1: Topics coherence bot (v1.0) was able to provide conversation drivers for. These were selected based on various hot topics in the news and tv at that time. Full list of drivers is presented in Appendix A.

In this version of the system, CoherenceBot is using a fixed handcrafted policy to select the action it will take on each turn. If the conversation is on a topic that the bot can handle, it would select a driver under this topic category. Otherwise, it would try to pick a topic based on the user preferences (if one exists and it can be handle by the bot). If no user preferences have been collected already, the *RapportBot* is forcefully called from within the CoherenceBot and follows the functionality as previously described in section 3.3.3.10. Finally, if none of these steps generate an appropriate response, the bot would prompt the user to pick a subject they are interested in (e.g. *"So, would you prefer to talk about music, celebrities, or movies?"*) from the predetermined list of topics the bot can handle (Table 3.1).

3.3.3.12 Seq2seq bot

During the initial phase of the competition, an attempt was made to introduce a Seq2Seq (Cho et al., 2014, Vinyals and Le, 2015) neural bot in the ensemble. This bot was trained on different open-source datasets (OpenSubtitles (Lison and Tiedemann, 2016), and Cornell Movie Dataset (Danescu-Niculescu-Mizil and Lee, 2011)) as well as Twitter data[18]. Although the generated responses were mostly

[18] https://github.com/Marsan-Ma/chat_corpus

grammatically correct, they could be offensive or politically related. This lead to the decision to quickly remove this bot from the deployed version of the system.

Problematic example responses of the seq2seq bot	
U:	Shall I kill myself?
S:	*(Seq2SeqBot)* Yes. . . .
U:	I am feeling down today.
S:	*(Seq2SeqBot)* You will die.

3.3.4 Selection strategy and Post-Processing

Once all the candidate responses have been gathered in the Bucket, they undergo post-processing and then a selection strategy is applied on the list to select the most appropriate candidate in the current turn.

Candidate responses containing profanities (which was strictly checked as the interaction had to be safe for children) were filtered out from the list. This task, however, was challenging, since inappropriate responses do not necessarily contain obscenities and are often context-dependant (Table 3.2). For example, the system's response *"I can sleep with as many people as I want"* (actual example generated by the experimental seq2seq bot described in section 3.3.3) does not contain any profane words that can be identified by a lexicon-driven approach, but is still inappropriate for a children-safe deployable system (e.g. U: *"Is Santa real?", S: "Santa is a fictional character traditionally associated with [...]")*.

Additionally, single-word candidates and repetitive responses (that were provided in the last 2 previous turns) were also filtered out since they would be less likely to progress the conversation any further and could make the interaction feel unnatural and boring or frustrating. Finally, responses containing references to inappropriate topics such as death, murder, rape, etc (from the information-retrieval bots) were also removed, as per the competition's requirements.

	User	**System**
Inappropriate language		You will die [*seq2seq trained on movies dataset*]
	<Asking about Santa>	Santa is Dead [*retrieved information from news article*]
	How do you dispose of a dead body?	With fava beans [*seq2seq trained on movies dataset*]
Unsafe counsel	Shall I kill myself?	Yes [*seq2seq trained on Twitter dataset*]
	What stocks should I invest in	Sell Sell Sell [*seq2seq trained on Twitter dataset*]

TABLE 3.2: Inappropriate examples collected from users interacting with the Alana system

The selection of the appropriate response from the (now filtered) list was then done in three steps:

1. **Bot priority list.** By analysing the conversation logs during the competition, the team was able to empirically prioritise the bots in the ensemble in terms of responses. Additionally, some of the more specialised bots where their functionality was restricted to a specific task (e.g. Quiz Game, WeatherBot, FactBot) took the highest priority. This priority list was fine-tuned several times during the competition depending on the user ratings over several 5-day long iterations. The final deployed priority list was: *RapportBot* $\rightarrow$ *IntroBot* $\rightarrow$ *QuizGame* $\rightarrow$ *FactBot* $\rightarrow$ *WeatherBot* $\rightarrow$ *PersonaBot* $\rightarrow$ *EviBot*.

   The remaining bots in the ensemble (i.e. WikiBot, NewsBot, ElizaBot, and CoherenceBot) all shared the same lowest selection priority.

2. **Contextual priority.** The next step was applied specifically to the NewsBot, where it takes priority in the current turn if the user was explicitly asked for confirmation to continue with an article or not in the immediate previous turn (as shown in the example in Section 3.3.3.3).

3. **Ranking function.** Finally, if none of the bots in the priority list produced an acceptable answer, a ranking function is used to select a response generated from the remaining bots. Details on the functions used are described in section 3.4.

4. **Fallback.** In the case that all of the bots in the ensemble failed to produce a suitable candidate, the FactBot is forcefully called by the Bucket to return a random fun fact. This fallback step was mandatory as the system had to always return a response to the user to keep the session active. Closer to the end of the 2017 competition, calls to the CoherenceBot were added to the fallback strategy. Thus the fallback response at by the end of the competition was selected as follows: $P_{cb} = 70\%, P_{ff} = 20\%, P_j = 10\%$, where cb denotes response from CoherenceBot, ff a fun fact from the FactBot, and j a joke from the FactBot.

Finally, the selected response was post-processed before being output to the user. In 2017 this was done in two steps: Any bot that wanted to use the user's name as part of their response, could inject the special tag `*username*` in the response. This tag was then replaced during the post-processing with the actual user's name from the database. Similarly, if a bot in the ensemble wanted to make use of a conversational driver generated by CoherenceBot, the special tag `*driver*` was used. In this case, the post-processing function called CoherenceBot instructing it to generate a driver for the given topic.

3.4 Ranker

As described in section 3.3.2, the list of candidate responses is ranked so the most appropriate response from the pool is selected before being post-processed and returned to the user. During the competition several different ranking functions were tested.

3.4.1 Hand-Crafted ranker

In this version of the ranker, several features of the dialogue are weighted differently (for instance, any NE, if they exist, are being boosted), in order to select the best candidate amongst the response pool. The hand-crafted ranking function uses the following features:

- **Coherence:** Inspired by Li et al. (2016c), semantic similarity is calculated between the user's utterance and each candidate using Word2Vec (Mikolov et al., 2013b) in order to maintain the overall topic of the conversation. Higher semantic similarity is rewarded.

- **Flow:** To prevent repetitive responses, similarity between consecutive system utterances is penalised.

- **Questions:** As one of the goals is to maximise engagement with the users, questions in the responses are also rewarded to encourage the user to continue the conversation.

- **Named Entities:** Responses containing NE also mentioned by the user are strongly rewarded, as this increases the probability that the information returned will be more relevant to the topic discussed.

- **Noun Phrases:** Similarly, matching NPs between the user's and the system's utterances are rewarded.

- **Dullness:** Again following Li et al. (2016c), dull responses such as *"I don't know"* are penalised. Word2Vec similarity between a list of dull responses and the user's utterance is used to identify these.

- **Topic Divergence:** For every candidate response in the Bucket the topic divergence from the user's utterance is calculated using a Latent Dirichlet Allocation (LDA)(Hoffman et al., 2010) model. This model was trained on a weighted combination of the OpenSubtitles and the WashingtonPost datasets.

- **Sentiment Polarity:** Finally, sentiment tagging was performed on the user's utterance using the VADER sentiment analyser (Gilbert and Hutto, 2014) from the NLTK toolkit,[19].

The weights were manually adjusted based on sample conversations with the final score being a weighted sum of these features (as shown in Papaioannou et al. (2017a)):

$$\begin{aligned} score = 0.25 * turn_0 + 0.25 * turn_1 + 0.25 * turn_2 + 0.25 * noun\_phrases \\ +3 * named\_entities - 0.25 * topic\_divergence \end{aligned} \tag{3.1}$$

where $turn_i$ denotes the i-th utterance from the end of the dialogue history:

$$\begin{aligned} turn_i = -0.2 * flow_{sem\_similarity} - 3 * flow_{METEOR} + 0.1 * coherence_{sem\_similarity} \\ -0.24 * dullness + 0.2 * question + 0.1 * sentiment\_polarity \end{aligned} \tag{3.2}$$

3.4.2 Linear Ranker

Additionally, a linear ranker was trained by Shalyminov et al. (2018), based on the VowpalWabbit linear model[20] (Agarwal et al., 2014). The ranker is trained using a Mean Square Error (MSE) loss function with features:

- bag-of-n-grams (1, 2, 3-grams) using the 3 previous utterances (SYSTEM-USER-SYSTEM) and the response
- n-grams that are position-specific at the start of the context and in the response (first 5 positions)
- dialogue flow features, as described in Section 3.4.1
- the bot's name

[19] http://www.nltk.org/api/nltk.sentiment.html
[20] https://vowpalwabbit.org/index.html

Although the ranker was initially trained using open-sourced datasets (Cornell movies (Danescu-Niculescu-Mizil and Lee, 2011), Twitter, and Jabberwacky[21]), its performance was rather poor as it was found that it strongly preferred responses similar to examples found in these datasets. Therefore, the ranker was retrained using real conversations from the system's interaction with its users during the competition. User ratings of 4+ were used as positive examples, while ratings of 1-2 were scored as negative. The challenge in this case was the fact that the reward signal given by the users was extremely sparse (only a single rating at the end of the whole dialogue) making it very unreliable to propagate this signal back to each turn in the conversation.

This ranker achieved 69.40% accuracy in classifying the development set (over 7,000 dialogues gathered during the competition).

3.4.3 Comparison of the rankers

Both versions of the ranker were evaluated during the competition, with the results presented in Table 3.3.

System	average user rating	number of dialogues
Alana v1.1 : Hand-engineered Ranker	3.26	191
Alana v1.1 : Trained Linear Ranker	**3.28**	272

Table 3.3: Results: Hand-Engineered vs Trained Linear Ranker

Although both rankers performed similarly in terms of user ratings, it was observed that even given sparse feedback the trained ranker performed as well as the hand-engineered one, and there is a trend showing that retraining with additional user data would potentially further improve the performance.

[21] http://www.jabberwacky.com/

3.5 Conclusion

This chapter presented an initial proposed solution to tackle the challenges of open-domain conversation, called *Alana*. The system was designed, implemented, and evaluated as part of a team project during the inaugural Amazon Alexa Prize competition between 2017-2018. The proposed solution includes an ensemble of different bots generating candidate responses and a trained ranker for selecting the most appropriate response in context. All the relevant modules in the architecture were discussed, including the NLU, DM, (both delivered by the centralised module called "Bucket") and NLG (in the form of a collection of different bots) capabilities of the proposed system.

Additionally, methodologies for tackling the challenges of Coherence and Engagement in open-domain dialogues were discussed (in the form of *IntroBot, RapportBot,* and *CoherenceBot*).

Finally, discussion on the system's DM in the form of a number of selection strategy steps and 2 versions of ranking functions were described, with initial evaluation results of the ranking functions presented.

Although the architecture followed and decisions made in this version of the system performed well, it was found that the system had several limitations. First and foremost, it was found that the way the pipeline was designed was quite rigid, making future extensions of the system hard to implement and integrate. Additionally, the system had very limited to no multi-turn capabilities, leading to very diverse but quite shallow conversations (high breadth but low depth in conversation topics). Furthermore, some of the architectural design decisions followed created unnecessary load to the system, leading to occasional system time-outs under heavy load. For example, WikiBot and NewsBot were occasionally re-annotating the user's utterance (e.g. POS tagging and NE extraction) although this information could be delivered by the Bucket as part of the NLU actions. Some of these limitations were tackled in the next version of the system, presented in Chapter 5.

In the next chapter, more in-depth analysis of the data gathered during the deployed period of this system (here referred to as *AL2017)* will be presented and further discussed.

Chapter 4

Alana 2017 Dataset Analysis

The aim of the work presented in this chapter is to explore possible relationships in the data collected during the Amazon Alexa Prize 2017 competition and identify which features can be further exploited to increase the overall quality of the conversation. This will enable the optimisation of the system and the **Research Questions 2** and **4** (see Section 1.2) to tackle the challenges presented in Chapter 1.

4.1 Dataset description

The Alana2017 dataset (henceforth called AL2017) consist of all interactions between the users and the system described in Chapter 3 between 01-07-2017 and 25-11-2017. During that period 65,667 distinct dialogues (sessions) were collected, summing up to a total of 2,354,937 turns (1,178,742 user and 1,176,195 system turns). The slight difference between user and system turns count is attributed to occasional system time-outs or other deployment faults, resulting in an empty system utterance. During that 2017 competition period (semi-finals + finals), the system managed to score on average at 3.60 on a 5-point Likert scale. The score was given at the end of the conversation and reflected the dialogue as a whole.

Some further descriptive statistics are shown in Table 4.1, with an additional breakdown between user and system utterance statistics shown in Table 4.2.

	rating	**num of turns**	**utterance length (user)**	**utterance length (system)**	**vocabulary size (user)**	**vocabulary size (system)**
mean	**3.60**	**35.56**	**3.79**	**21.91**	**3.76**	**20.60**
std	1.37	42.34	3.99	15.39	2.29	8.07
min	1.00	2.00	1.00	1.00	1.00	2.00
25%	3.00	10.00	1.00	12.00	2.40	16.00
50%	4.00	22.00	3.00	18.00	3.33	20.50
75%	5.00	46.00	5.00	29.00	4.56	24.45
max	5.00	906	105	1,735	73	591.33

Table 4.1: AL2017 aggregated statistics

num of dialogues	65,667
num of turns	2,354,937
user turns	1,178,742
system turns	1,176,195
vocabulary size (user)	48,448
vocabulary size (system)	190,315
avg user utterance length	3.79
avg system utterance length	21.91

Table 4.2: AL2017 Overall statistics. The difference in the number between user and system turns is due to random crashes of the system during the competition which failed to produce a response.

The user vocabulary includes 48,448 distinct words across all sessions, whereas the system vocabulary includes 190,315 words. The system vocabulary includes tokens present in the responses' templates as well as those included in retrieved information (e.g. from the NewsBot and WikiBot). It is worth mentioning that the extreme maximum values noted in number of turns, and user and system utterance length (906, 105, and 1,735 respectively) are considered outliers. For example, the extreme value of system utterance length is attributed to an oversight in the development of one of the bots that wasn't properly filtering the information retrieved.

In Table 4.3 the distribution of bot usage is observed, in terms of times triggered across all sessions as well as the percentage it holds across all system responses (1,176,195).

	Num of responses	%
PersonaBot	212,082	18.03
ElizaBot	193,716	16.46
QuizGame	181,324	15.41
NewsBot	181,222	15.40
EviBot	81,773	6.95
FactBot	73,422	6.24
IntroBot	70,097	5.95
CoherenceBot	73,465	4.94
WikiBot	42,150	3.58
RapportBot	21,038	1.78
WeatherBot	851	0.07

TABLE 4.3: Distribution of bot usage. The third column (%) denotes the percentage of that bot over all of the system's turns in the corpus

As shown in Figure 4.1, the system's rating started off rather low but progressively got better over the course of the competition as new features were incrementally added to the system. Additionally, daily inspection of the user interaction logs provided the necessary feedback and insight of the system's performance "in the wild", and a better understanding on how to model the user's language and intentions when presented with an open-domain system.

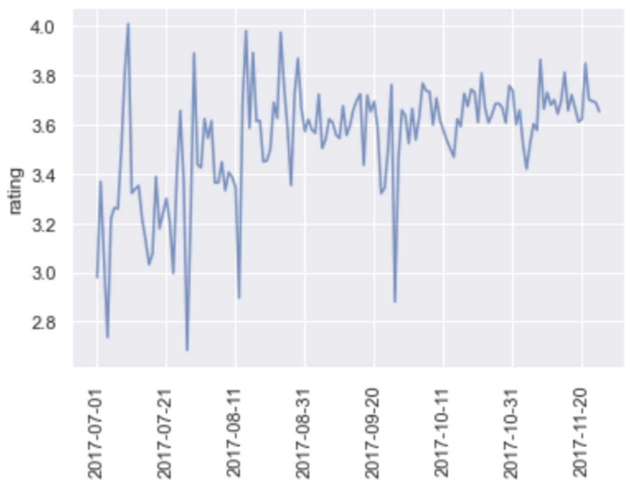


FIGURE 4.1: Rating progression during the 2017 competition.

The various sudden dips in ratings (e.g. end of July, end of September, etc) are attributed to rare unforeseen errors of the NLU which led to no annotations produced. Additionally, throughout the competition, Amazon released newsletters inviting the general public to interact with the socialbots leading to a very high number of daily calls to the system. Finally, at the end of September, Amazon developers were invited to interact with the socialbots to test their performance and capabilities. This resulted in targeted scoring given the developers' background knowledge of the scope of the competition.

4.2 Correlation between dialogue length and rating

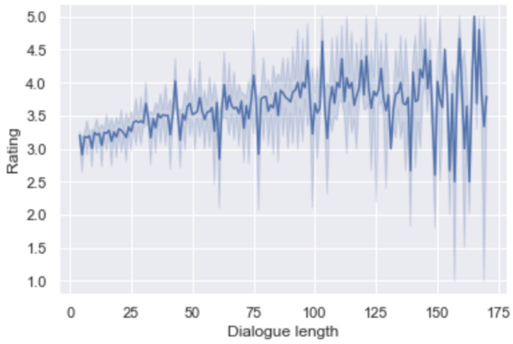


Figure 4.2: Relationship between the average dialogue length (number of turns) per session. The faded areas denote aggregation over multiple y values at each value of x and shows an estimate of the central tendency and a confidence interval for that estimate.

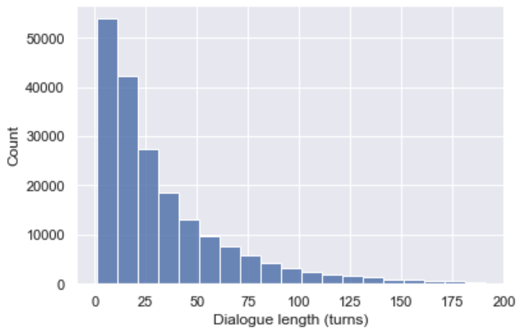


Figure 4.3: Distribution of average dialogue length per session.

To investigate whether the length of the dialogue has any effect on the dialogue's rating, an initial analysis of the data is needed in terms of the users' utterances. As mentioned earlier, the AL2017 dataset includes 65,667 dialogues with an average length of 35.56 turns per session ($\sigma = 42.34$), minimum of 2 turns, and maximum length of 906 turns per dialogue. As also shown in the distribution of the dialogue's length in Figure 4.3 it was decided that very long conversations ($\geq$ 174 turns - $\pm$3.29[1] standard deviations from the mean) and very short ones ($\leq$ 4 turns - dialogues with only 2 full turns, i.e. user + system, were considered not to have enough context to draw conclusions from) will be discarded as outliers.

[1]As Tabachnick and Fidell (2013) suggests, values beyond $\geq$ 3.29 standard deviations from the mean are indicative of an outlier.

Following this, the correlation between the dialogue's length (independent variable) and rating (dependent variable) was calculated. Since the length is a continuous value whereas the rating is the result of a 5-point Likert scale, thus can be considered ordinal, both Spearman's rank correlation (ρ) and Kendall's coefficient (τ) can be used (Khamis, 2008). The correlations found were: Spearman ($\rho = 0.1192$), Kendall ($\tau = 0.0898$) showing that no strong correlation exists between these two variables. It is evident however from these values and Figure 4.2 that a weak positive correlation exists, although it is also observed that lengthier dialogues result in more diverse ratings.

Please note that given that the analysis was performed on the entire dataset (excluding outliers) and not on a specific sub-sample (since the entire dataset is known and annotated). This means that given the large sample size, any significance calculations would most likely provide a significant p-value (Hole, 2014, Murtaugh, 2014).

4.3 Correlation between user/system utterance length and rating

To investigate the effect of the length of both the user and the system utterances on the dialogue's rating, a similar approach to the previous section was followed.

The AL2017 dataset includes roughly 1.2M user utterances (turns) with an average length of 3.79, minimum of 1, and maximum length of 105 ($\sigma = 3.99$) words per turn (Table 4.1). The outliers were set as user turns longer than 13 words (± 3.29 standard deviations from the mean) as also supported by the histogram in Figure 4.4. The remaining sample includes 99.90% of the user utterances.

Similarly, the length of the average system utterance length was 21.91 ($\sigma = 15.39$) tokens with a minimum of 1 and maximum of 1,735. The outlier threshold was taken again ± 3.29 std from the mean at 73 words length (keeping 99.97% of the system utterances).

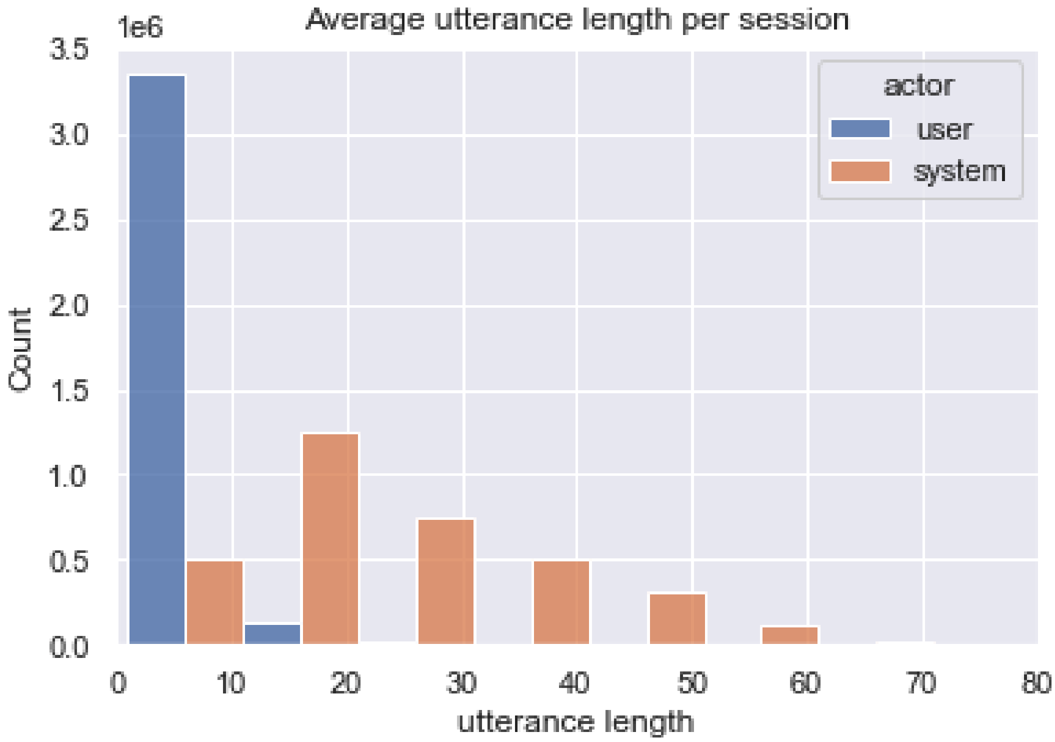


FIGURE 4.4: User and system avg turn length distribution

To calculate the correlation, the length of the average user turn of each session is calculated and compared against that dialogue's rating. The results are summarised on Figure 4.5.

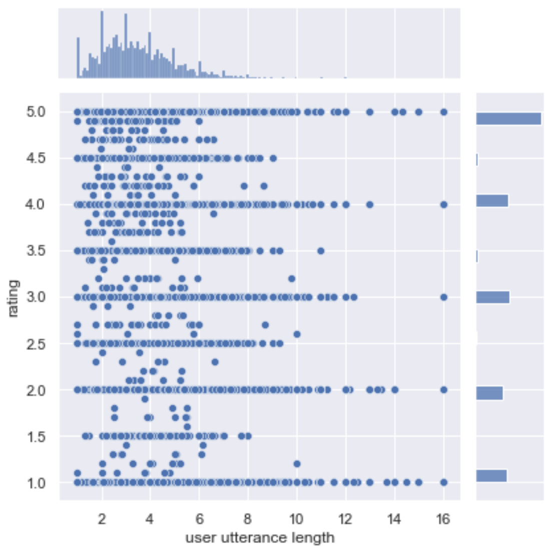


FIGURE 4.5: User avg turn length scatter plot with additional distributions of ratings and length.

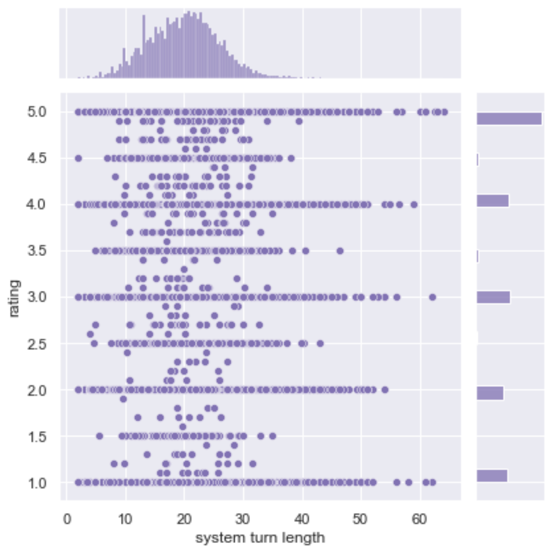


FIGURE 4.6: System avg turn length scatter plot with additional distributions of ratings and length.

	Spearman-ρ	Kendal-τ
User	-0.0902	-0.0669
System	0.1021	0.0751

TABLE 4.4: Correlations between user and system utterance length and dialogue ratings.

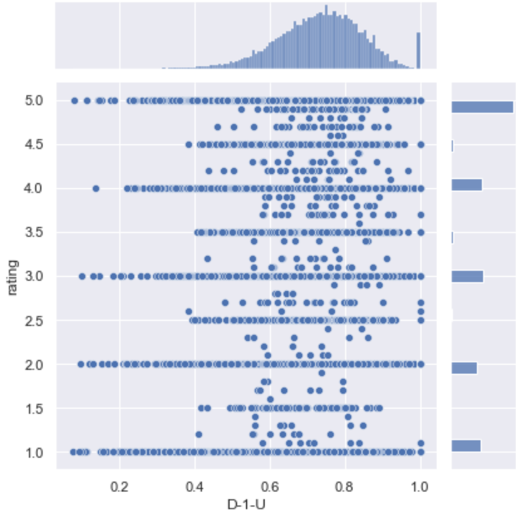
As presented on Table 4.4, no notable correlation is found between the length and rating of the user turn, however a slight trend of *negative* correlation for the user's utterance length can be observed indicating that longer user turns lead to lower ratings. These observations will be discussed further at the end of this chapter.

Additionally, a weak *positive* correlation between the *system* turn's length and the dialogue rating is observed.

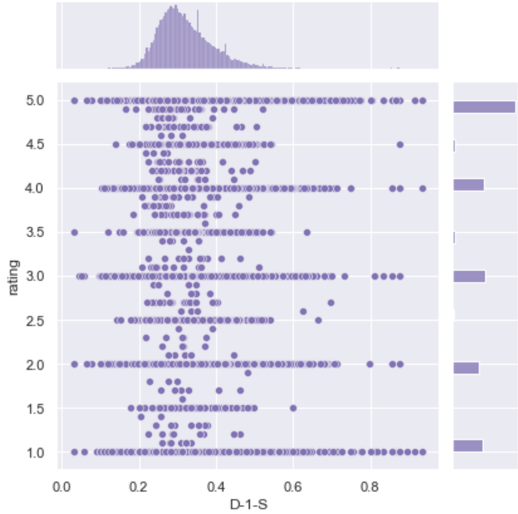
4.4 Correlation between lexical diversity and rating

This correlation study focuses on whether some relationship exists between the user and system lexical diversity and the dialogue ratings. Thus, the distinct n-grams metric (Li et al., 2016b) is employed, which measures the diversity of a sentence. It penalises sentences with a high number of repeated words as it is concerned with the number of distinct n-grams in a sentence. The metric is devoid of any reference or ground truth sentence and is entirely focused on the property of a sentence. Based on the work presented by (Dušek et al., 2020) the Distinct-N metric can also be used to measure lexical diversity.

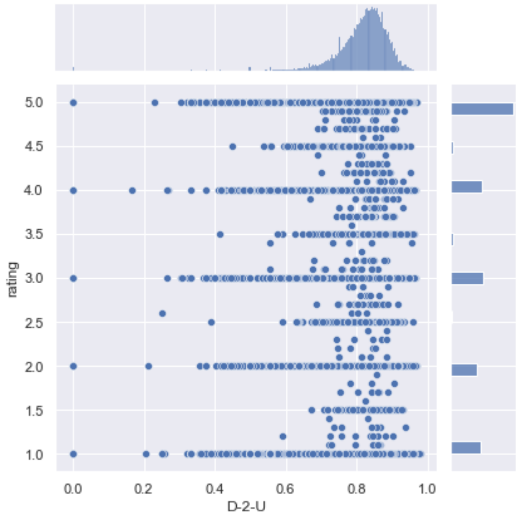
Using this metric, the unique uni- and bi-grams (D-1 and D-2 respectively) were calculated on the summed pool of all user and all system utterances, which are summarised on Table 4.5 and Figures 4.7. Given that the conversations rely heavily on NEs (both as part of the system's responses as well as be provided by the user), D-2 also helps capturing those entities as a single n-gram.



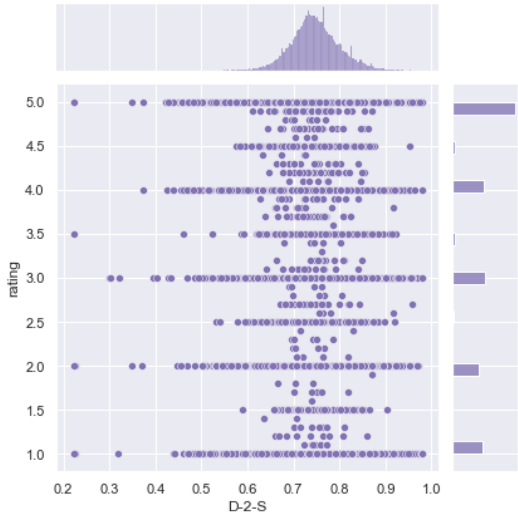
(A) Distinct uni-gram across all *user* utterances



(B) Distinct uni-gram across all *system* utterances



(C) Distinct bi-gram across all *user* utterances



(D) Distinct bi-gram across all *system* utterances

FIGURE 4.7: D-1 and D-2 distributions over all user and system utterances

Following that, the correlations between the average D-1 and D-2 for both the user and the system against the dialogues rating were calculated (Table 4.5).

Based on the findings outlined in Table 4.5 there is no correlation between the lexical diversity of either the user nor the system and the dialogue rating.

A clearer comparison between the lexical diversity between the user and the system can be observed in Figures 4.8(A) and (B).

		D-N	Spearman-ρ	Kendal-τ
1-gram	user	0.73	0.0992	0.0735
	system	0.31	-0.0901	-0.0661
2-gram	user	0.81	-0.0526	-0.0390
	system	0.74	-0.0748	-0.0549

TABLE 4.5: Average Distinct-N on corpus level for both user and system.

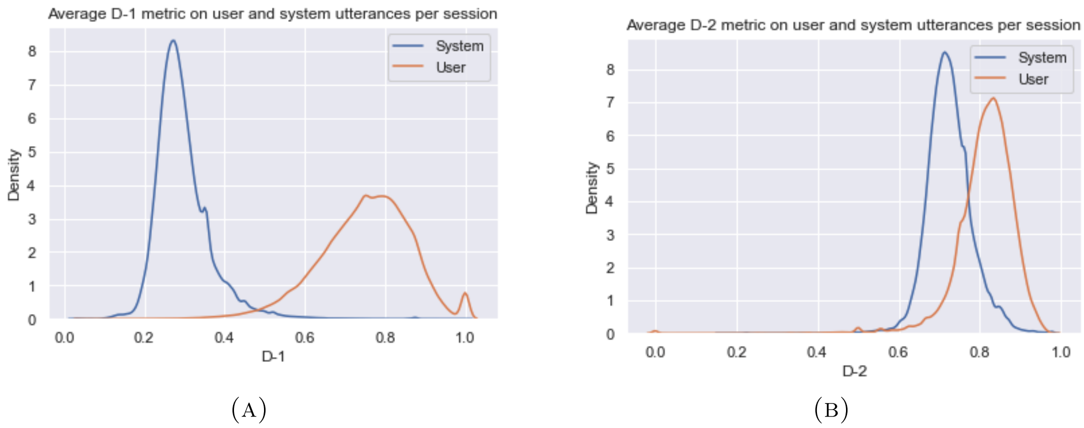


FIGURE 4.8: Comparison between user and system lexical diversity based on 1- and 2-grams

4.5 Named Entities Analysis

	NEs per turn		**NEs per session**	
	User	**System**	**User**	**System**
mean	0.20	1.51	3.60	32.92
std	0.24	2.29	5.39	51.45
min	0.00	0.00	0.00	0.00
25%	0.00	0.50	0.00	3.00
50%	0.15	1.31	2.00	15.00
75%	0.28	2.19	5.00	41.00
max	8.00	317.66	138.00	1,071.00

TABLE 4.6: Named Entities per turn and per session mentioned by the user and by the system across all turns in the corpus.

Given the architectural decisions discussed in Section 3.3.3 regarding the usage of NEs as the driving force of the flow of conversation (especially the effect of NEs that include a PERSON, LOCATION, or ORGANISATION), a question is raised about the effect of the NEs on the quality of the conversation (rating).

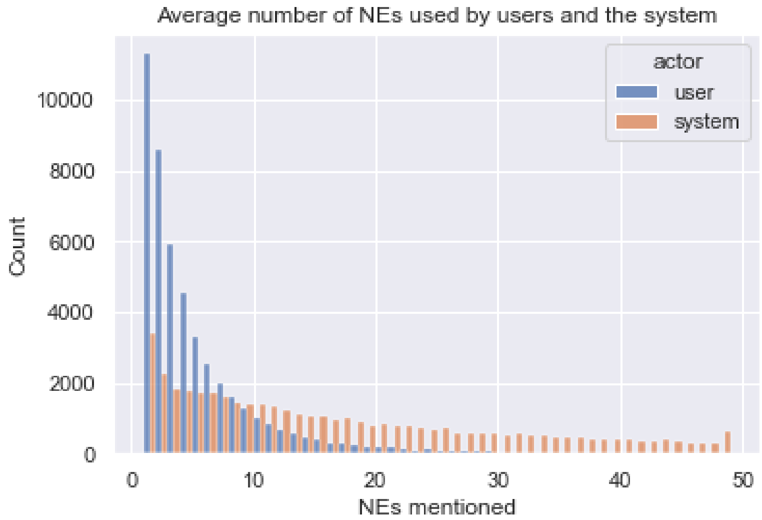


FIGURE 4.9: Average Named Entity distribution *per session* for both user and system. Users tend to use a very small amount of NEs in their utterances while the system includes a more normally distributed number.

To investigate this relation, every user and system utterance was parsed using the spaCy EntityRecogniser[2], specifically the *en_core_web_sm* model, to extract all NEs from the text. The model is able to identify entities that fall under the categories CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP[3], ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, and WORK_OF_ART. A subset was selected from this list which excludes the entities of type DATE, CARDINAL, and ORDINAL, since the primary focus of this section is on the effect of entities that the system can map into topics, such as names of persons, etc.

Similarly to the previous sections, based on the statistics presented in Table 4.6 and the histogram in Figure 4.9, the dataset includes outliers that need to be identified and removed. The users use a relatively small number of NEs in their utterances – average 0.20 ($\sigma = 0.24$) per turn and an average of 3.6 per session. The system uses a considerably larger amount of NEs, primarily due to the information retrieval bots (i.e. NewsBot, WikiBot, and EviBot) whose responses may include a number of different entities. The extreme maximum value shown in Table 4.6 for the system is due to the fact that for a small period of time for some of these

[2]https://spacy.io/api/entityrecognizer
[3]NORP: Nationalities or religious or political groups

	Spearman-ρ	Kendal-τ
user	0.1102	0.0863
system	0.1317	0.0994

TABLE 4.7: Correlation between max number of Named Entities mentioned per session and dialogue ratings.

bots (such as EviBot) no limit on the response's length was put into place, leading to responses with a considerably large amount of NEs. The outlier threshold was selected as in the previous sections at conversations that the user was using up to 22 NEs per session and the system up to 200 as also visualised in Figure 4.9 (x-axis truncated at 100 for better readability). This sample still included 99.60% of the sessions in the corpus.

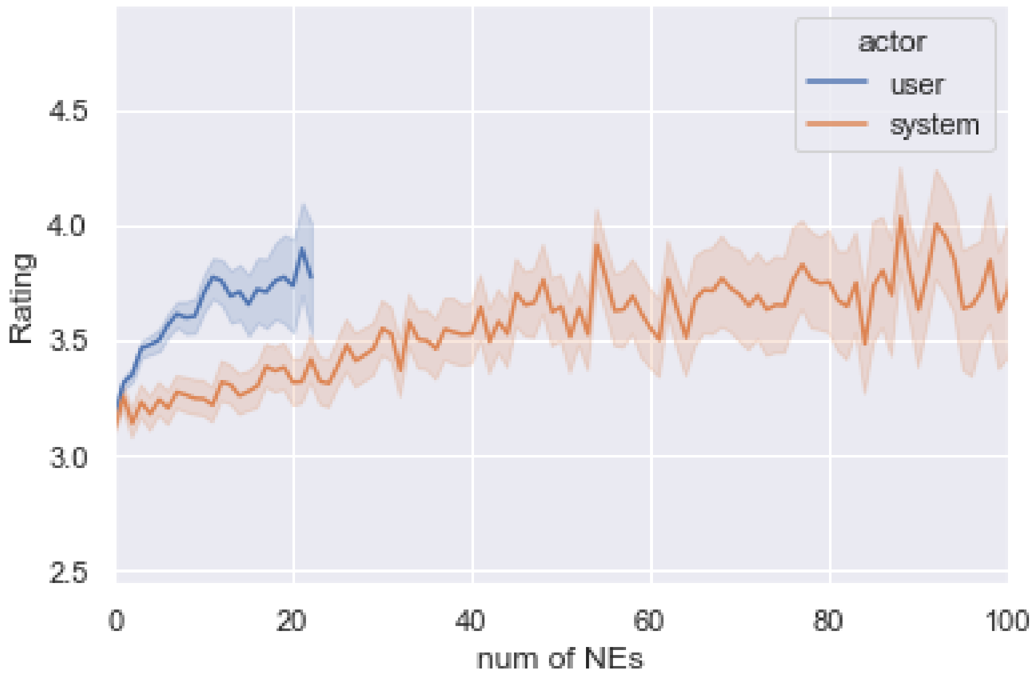


FIGURE 4.10: Relation between number of NEs and dialogue rating. Axis x shows the total number of NEs per session mentioned by the user and the system. The faded areas denote aggregation over multiple values on the y axis at each value of x and shows an estimate of the central tendency.

Again, given the continuous-ordinal nature of the number of NEs and rating scores respectively, both Spearman and Kendall correlations were calculated as presented in Table 4.7. A weak positive correlation is observed between the number of NEs produced (by both the system and the user) and the rating, meaning that users enjoyed more when the system was mentioning multiple Named Entities throughout the course of the conversation, possibly because in an open-domain environment this was giving the user the necessary triggers to continue an enjoyable conversation. Conversely, the correlation between the number of NEs in the user's utterances shows a slightly weaker positive correlation with the dialogue's rating up to about 10 entities per session, however after that point the central tendency in the data appears to remain unchanged, as shown in Figure 4.10.

4.6 Hypothesis testing of bot significance

Given the importance of NEs in the overall architecture, further questions are raised with regards to the importance of the presence of the two bots primarily responsible for using NEs as part of their responses, NewsBot and WikiBot. Thus, the formulated null hypotheses are:

- $H0_{1a}$: **NewsBot has no effect on the dialogue ratings**
- $H0_{1b}$: **WikiBot has no effect on the dialogue ratings**

Following the same rationale, it is also interesting to investigate the effect that few of the top responding bots in the ensemble (as shown in Table 4.3) have on the overall quality of the dialogue, adding the following hypotheses:

- $H0_{1c}$: **ElizaBot has no effect on the dialogue ratings**
- $H0_{1d}$: **PersonaBot has no effect on the dialogue ratings**

To investigate these hypotheses, independent samples were selected from the corpus in the following way: For each bot (WikiBot, NewsBot, ElizaBot, PersonaBot) a sample of $N_1 = 1000$ dialogues was selected where the bot is present in the responses ($S_{wiki_1}, S_{news_1}, S_{eliza_1}, S_{person_1}$) as well as a sample of ($N_2 = 1000$) sessions that *do not* include any responses from that particular bot ($S_{wiki_0}, S_{news_0}, S_{eliza_0}, S_{person_0}$).

Judging by the samples' Shapiro test (Shapiro and Wilk, 1965)[4] results ($S_{wiki_1} = 0.84, S_{news_1} = 0.85, S_{eliza_1} = 0.87, S_{person_1} = 0.86, S_{wiki_0} = S_{news_0} = 0.85, S_{eliza_0} = 0.85, S_{person_0} = 0.84$) none of these samples are normally distributed. Furthermore, the rating (dependent value) is of ordinal level. Thus, to test the significance of each bot to the ratings, the Mann-Whitney-U test is appropriate, which compares differences between two independent groups (existence or not of a bot) when the dependent variable (rating) is either ordinal or continuous, but not normally distributed[5].

Given the sample size being large ($N > 20$) the normal critical value table for significance (see Appendix B) can't be used so U is approximately normally distributed and the significance calculated using the formula 4.1, where where mU and σU are the mean and standard deviation of U.

$$z = \frac{U - m_U}{\sigma_U} \tag{4.1}$$

The results are presented in Table 4.8. Three out of the four tested bots show that the difference between a randomly selected value of Sample1 and the Sample2 populations for each pair is big enough to be statistically significant ($p < 0.05$). ElizaBot's presence however has no significant impact on the rating ($p = 0.38$), however, all bots individually had a positive effect on the ratings. Thus, hypotheses $H0_{1a}, H0_{1b}$, and $H0_{1d}$ can be rejected while $H0_{1c}$ cannot.

These results, however, raised an additional question regarding the correlation between the number of times each bot fired within a dialogue and the dialogue's rating. Thus, the Kendall and Spearman correlations were calculated as shown in Table 4.9.

From the four tested bots, WikiBot, ElizaBot, and PersonaBot show positive correlation with the dialogue's rating. NewsBot shows a negligible positive correlation

[4]The Shapiro– Wilk test tests whether the the data are independent and identically distributed given a sample X1,..., Xn of n real-valued observations

[5]https://statistics.laerd.com/spss-tutorials/mann-whitney-u-test-using-spss-statistics.php

	sample average rating	sample std	sample media	sample size	Z	p-value
S_{wiki_1}	3.62	1.46	4	300	3.05	**0.0016**
S_{wiki_0}	3.21	1.55	3	300		
S_{news_1}	3.76	1.34	4	300	4.26	**1.0753e-05**
S_{news_0}	3.21	1.55	3	300		
S_{eliza_1}	3.45	1.41	4	1000	0.84	0.3847
S_{eliza_0}	3.36	1.51	4	1000		
$S_{persona_1}$	3.57	1.37	4	1000	3.02	**0.0018**
$S_{persona_0}$	3.32	1.53	4	1000		

Table 4.8: Two-tailed Mann-Whitney U test on the 2017 dataset. Due to the large sample sizes the significance Z is calculated using the formula 4.1.

	Spearman-ρ	**Kendal-τ**
WikiBot	0.1668	0.1299
NewsBot	0.0084	0.0902
ElizaBot	0.1396	0.1379
PersonaBot	0.1627	0.1235

Table 4.9: Correlation between the number of responses per session of each bot and the dialogue's rating

however, indicating that more news delivered in a single sessions does not necessarily have a positive impact on the user's engagement.

4.7 Observation and Discussion

In this chapter the data collected during the semifinals and finals of the Alexa Prize 2017 (named AL2017 dataset) was analysed. The goal was to find any underlying relationships between the various features implemented in the system. Dialogues before the start of the semifinals stage and after the end of the finals were not included. This was due to the fact that at the very beginning of the competition the team was still designing the core features of the system thus the number of confounding variables to measure was extremely high. Similarly, after the end of the competition the system was kept online by Amazon but no further maintenance was performed by our team resulting in overall poor performance due to frequent timeout errors and the usage of the system for a number of different research experiments performed by the individual members of the team.

One of the main challenges of this task was the fact that the users provided only a single rating at the end of each dialogue, making it hard to attribute that rating to specific features and phenomena present in the dialogue. It was very frequently observed that a dialogue for the most part could be quite engaging and coherent, but a slight mistake on the system's side or a controversial subject mentioned at the end of the conversation would lead to an overall low rating, thus masking any positive signal coming through for further evaluation. Additionally, no ASR information was collected in the 2017 competition resulting in extrinsic factors (such as very low ASR scored utterances) to be included in the dataset but not further examined.

Although most of the data analysis performed in this chapter showed trivial or no correlation between the various dependent variables (i.e. dialogue length, user/system length, number of NEs, etc), various trends are obvious that showcase the impact of said features to the overall performance of the system.

The first interesting outcome observed is that, as explained in Section 4.1, even though the system's NLU was occasionally completely broken, the users still were rating the interaction on a minimum average of ∼2.7-2.9 on the 5-point Likert scale. This shows the importance of carefully designed fallback strategies and NLG templates that enable at least partial recovery of the conversation.

As a general observation, the users varied a lot in how they rated the system. There were several occasions (confirmed by the team going through the transcript logs on a daily basis) where users would have a very interesting and engaging conversation, but give a very low rating in the end, and conversely, very short (1-, 2-turn) dialogues where they were rated highly. To a certain extent however this was to be expected, as the users were presented with the single question *"Would you like to interact with this system again in the future"*, without any context on the goals of the competition.

Regarding the findings about the correlation between user/system length and rating, a weak negative and weak positive correlations are observed from the users' and system's perspective respectively. This can be attributed to a variety of reasons: (a) Users that are not quite versed with speech technologies tend to formulate their utterances in a more human-like manner without consideration that they are not talking to an actual human, often combining multiple different intents and topics in the same utterance. Additionally, a number of linguistic phenomena that are not captured by the system such as self-corrections (e.g. *"Do you know anything about Space Track? No sorry I mean Jump. Space Jump"*) tend to increase the utterance's length, however, they are handled poorly by the system leading to lower scores. (b) On several occasions, the system picked up voice that wasn't intended for it (e.g. TV, radio) also leading to low scores. (c) Finally, a significant increase of interactions with younger-aged users was observed around holiday seasons. This led to language style that the system was not designed to handle, as well as random, uncorrelated ratings given to the system irrespective of the system's performance.

Regarding the impact of the number of NEs in the user/system utterance had on the ratings, a slight positive correlation is observed on Figure 4.7 and Table 4.7 on the system's side. Interestingly, a high variance in user's ratings was observed when it comes to *what type* of entities were mentioned by the system. Specifically, when the NewsBot was providing information on heavily polarised topics, such as political-oriented articles (e.g. articles about Donald Trump), the users rated the whole conversation poorly. Ultimately, it was decided by the team to completely filter out any related responses regarding such highly controversial entities.

No ASR information was collected during the 2017 competition so the confidence of the user utterance's quality could not be taken into account, a fact that was remedied in the 2018 Alexa Prize Challenge discussed in Chapters 5 and 6.

Equipped with the knowledge acquired in the 2017 competition, the team addressed a number of these challenges during the 2018 competition, described and discussed in the following chapter.

Chapter 5

The Alana v2.0 conversational framework

The success of the Amazon Alexa Prize Challenge 2017 competition lead the team to participate again in the next Amazon Challenge in 2018. The new team consisted again of 6 PhD students (Ioannis Papaioannou, Amanda Cercas Curry, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, and Xinnuo Xu) and 5 faculty advisors (Oliver Lemon, Ondřej Dušek, Arash Eshghi, Ioannis Konstas, and Verena Rieser). The team competed against 200 entrants, and won the 3rd place in that year's competition (Figure 5.1).

In this chapter the updated Alana system deployed in 2018 will be discussed, including the various changes implemented that aimed to increase overall system coherence and engagement with its users. Some of these features include:

- Better *multi-turn* dialogue state tracking
- Usage of Entity Linking to increase the flow of conversational topics
- Improvements to the NLU module
- Added bots in the ensemble (such as OntologyBot, RedditBot)
- User abusive behaviour detection

- User modelling

Initially, a brief description of the updated architecture is presented, followed by the design decisions made in the state tracking of the system, the new central component called the Hub, and a description of the consolidated NLU module. As in Chapter 3, all new and updated bots in the ensemble are described, as well as how the user's interests are being modelled and used in the pipeline.

The work presented in this chapter is the outcome of the joint efforts of the team during the Amazon Alexa Prize 2018. My work and contribution primarily focused on the system architectural design and optimisation, as well as the design, implementation, and evaluation of CoherenceBot, user modelling capabilities, and optimisation of the strategies used by CoherenceBot as described in this chapter.

In (Table 5.1), an example dialogue as produced by the 2018 system is presented, showcasing all the different features to be discussed in this chapter.

		[after four greeting turns...]	
5	**Alana:**	`Welcome back then USERNAME. Since I remember that you like movies, I am looking forward to the new movie Jurassic world that was just released. What movie are you looking forward to watching?`	
	User:	`Vanilla Sky` [→ "I'm looking forward to watching Vanilla Sky"]	Fragment resolution
6	**Alana:**	`Oh I heard something interesting about Vanilla Sky. In the cell, Carl Jung's book, Memories, Dreams, and Reflections can be seen on the table between David and McCabe. The book is all about Jung's personal dreams and how they helped him uncover his shadow and remove his persona or mask. Speaking of movies. What actors do you like? I like Leonardo dicaprio a lot. He was awesome in the Revenant.`	Movie Trivia
	User:	`Interesting`	

7	**Alana:** I just think the world is full of strange and unexpected things like that. So, talking about movies, What famous actor or actress would you like to meet? I would love to meet Will Smith. He's just so funny!	Persona + Coherence
	User: who is he?	Co-reference Resolution
8	**Alana:** I guess you know that Will Smith is an American actor, film producer and rapper. His best movies are Men in Black II, Enemy of the State, Men in Black 3, and Men in Black and he has won the Grammy Award for Best Rap Solo Performance. So USERNAME, on the topic of movies. Whats a movie that you like? I love Star Wars the Force Awakens. I think Chewbacca is my favourite character.	Movie Ontology: Entity explanation
	User: I want to talk about music	Topic navigation
9	**Alana:** So, talking about music. Who is a singer that you like? I really like Taylor Swift, because her voice is so beautiful.	Coherence
	User: Nina Simone [→ "I like Nina Simone"]	Fragment resolution
10	**Alana:** Yes. Someone was talking about Nina Simone the other day. Apparently, Daughter Lisa Celeste 'Lisa Simone Kelly' , born in 1962. She's a Broadway actress, star of the Broadway musical Elton John's Aida . Oh, okay. What is a musician that you like? I love Beyonce.	Music Ontology
	User: could we talk about something else?	Topic change request
11	**Alana:** So USERNAME, since I know you like books , what have you read recently? I am reading Diary of a wimpy kid.	User modelling

	User:	`Handmaid's Tale` [→ "I read Handmaid's Tale"]	Fragment resolution
12	**Alana:**	`Oh so you like to read a good science fiction? Maybe you know that the book Do Androids Dream of Electric Sheep? has the same genre. You might like that if you haven't already read it. What's another book like The Handmaid's Tale that you enjoyed?`	Book ontology: recommendation
	User:	`Let's talk about Tom Hardy`	
13	**Alana:**	`Oh I know tom hardy. Do you mean Tom Hardy the English actor, screenwriter and producer or Thomas Hardy the English novelist and poet?`	Clarification interaction
	User:	`I mean the novelist`	
14	**Alana:**	`Yes. Some friends told me something interesting about Thomas Hardy. He was an architect before becoming a full-time writer. So USERNAME, on the topic of movies. Who is a movie hero that you like to watch? I like Han Solo.`	Book ontology: trivia
	User	`What do you know about pizza?`	
15	**Alana**	`I was thinking that whenever a pizza commercial has stretchy melty cheese, it looks good and appealing, but when it happens in real life it's annoying and messy.`	Reddit response retrieval

TABLE 5.1: Example Dialogue with Alana 2018 as presented in Curry et al. (2018).

5.1 Overall System Architecture

In terms of the Challenge's goals and processes, they remained unchanged from the inaugural competition of 2017 as described in Chapter 3.1.

In terms of architectural changes, the team focused on addressing some of the major drawbacks of the 2017 architecture:

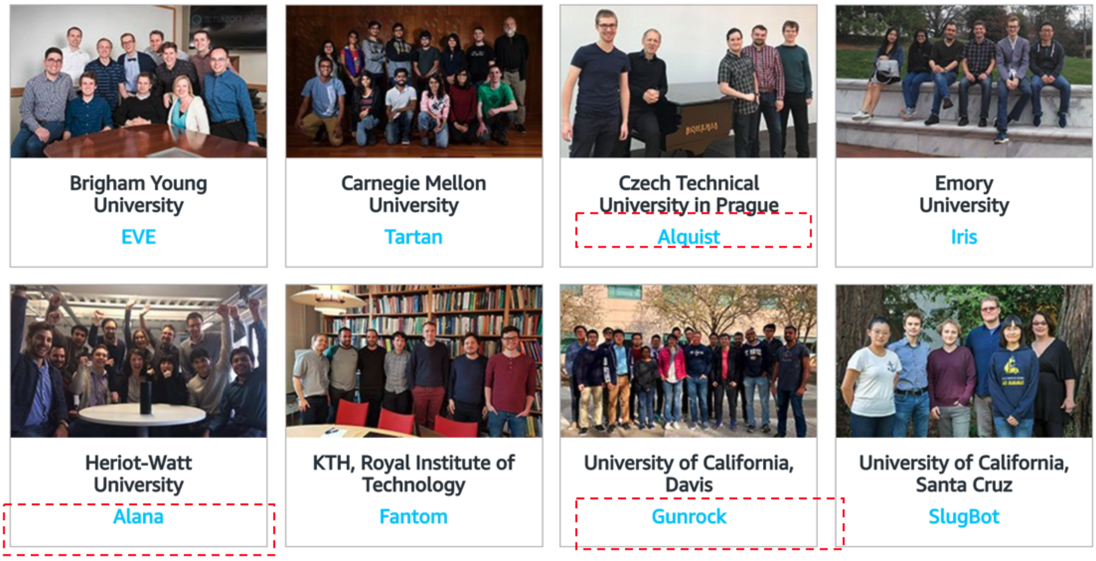


Figure 5.1: Alex Prize 2018 competing teams. 200 entrants, 8 semi-finalists, 3 finalists (denoted by dotted box).

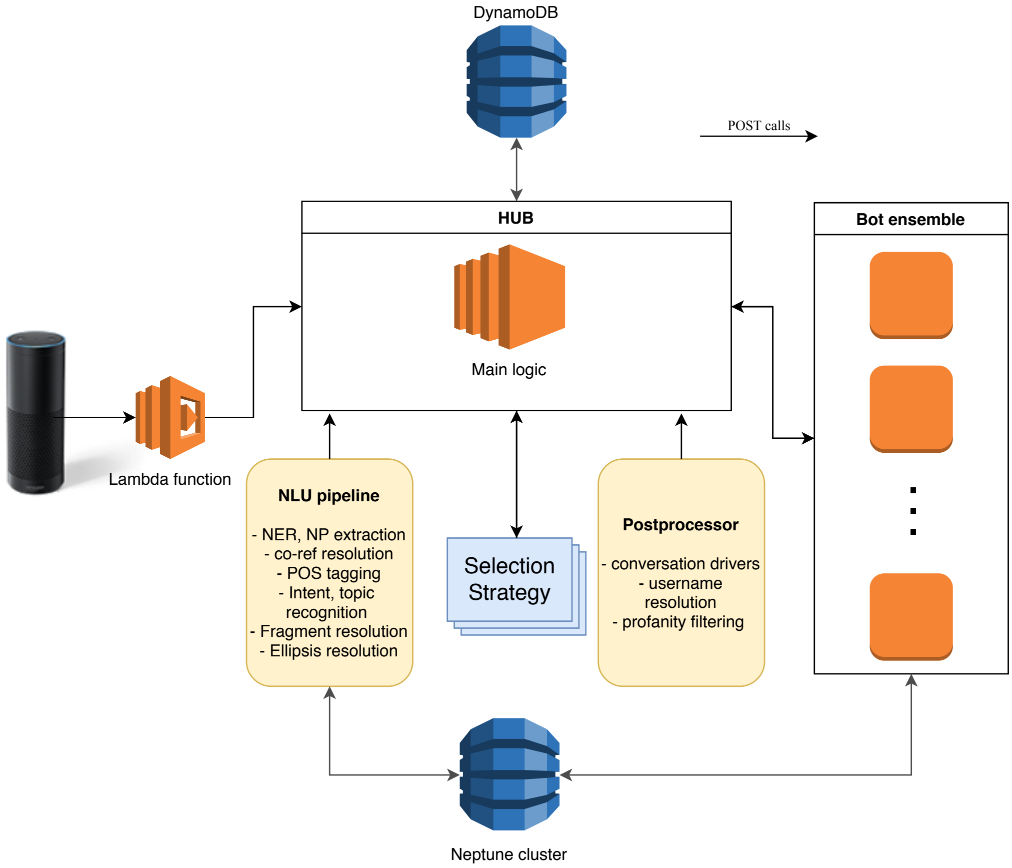


Figure 5.2: Alana 2018 architecture

- **System scalability:** The system had to scale better with the number of calls it was receiving on a daily basis.
- **Improved modularity:** A more modular architecture was required to accommodate the continuous maintenance and improvements Alana underwent during the competition.
- **Multi-turn conversation:** As the responses generated by the 2017 version of the system were primarily single-shot user-system turns, in 2018 the focus shifted into how to encode and carry over more context across turns throughout the conversation, through the use of better state tracking, improved NLU, and the use of Linked Entities.

The Alana 2018 modular design is shown in Figure 5.2, illustrating the general overhaul of the architecture in terms of process distribution and scalability. As we wanted Alana to be able to hold a more natural conversation across multiple turns, a mechanism for representing and sharing all useful information to be used as context throughout the pipeline was devised, through the implementation of the *state object*, as will be described in Section 5.1.1.

The overall pipeline of the system can be summarised as follows, and will be further described in the following sections: Like in v2017, when the user talks to the system, a new session begins and an event object reaches the *lambda function*, which contains several high-level metadata, such as user device's (Amazon Echo) ID and the text representation of the user's utterance from the ASR. The *lambda function* then forwards this information to an EC2 instance (henceforward called the *Hub*), which provides the main logic of the system.

This updated architecture and its motivation will now be presented in more detail.

5.1.1 The State Object

The state object is a custom Python object, represented as a dictionary, that encodes all the information shared across the different Alana sub-modules in the architecture. The model includes information about the current turn, including NLU annotations, candidate responses from the different bots, user information, etc (as shown in Figure 5.3). This centralised information allows additional modules (e.g. new bots) to extend Alana in a streamlined way, partly addressing the issue of rigidness of extendability of the 2017 system.

All sub-modules (including the bot ensemble) in the architecture have the same information available to them and they extract only the parts of the object that they require for their operation. This provides a more structured and robust representation of the conversation's context in a multi-turn scenario than the design followed in 2017.

Although most attributes stored in the state object are self-explanatory, a few notable attributes include the *processed text* (`state.nlu.processed_text`) which holds the user's utterance after the NLU pipeline has run (as explained in Section 5.2.2), which (in several cases) heavily alters the sentence compared to the raw text input which is stored in `state.input.text`. For instance, elliptical utterances (e.g. S: *"Do you like ice-cream?"*, U: *"Yes"* (elliptical turn)) would be stored under *state.input.text* whilst the resolved utterance (e.g. *"Yes I do like ice-cream"*) is stored in *state.nlu.processed_text*.

Another notable attribute is the *bot_states*. As explained in Section 5.1, since the Hub is the only module in the architecture that has access to the database, and given the ephemeral nature of each call to Alana (conversational turn), each bot needs a way to preserve their internal states across different turns. This is done via the *bot_states* state object attribute, which lets each bot store information in whatever structure they want (that can be hashed in a dictionary). Since all the bots are called simultaneously, and bots operate independently of each other, they all enrich a *copy* of the state object with their respective response candidates and internal state representations (as explained in more detail in sec: 5.2.3.1). In the response selection phase of the pipeline though only one bot's response and state is selected and saved in the database. This functionality allows the bots to keep

```
{
'session_id': ...,
'timestamp': ...,
'user_id': ...,
'last_state': {...}, # previous turn's state object
'state': {
   'last_bot': ..., # name of the bot that responded during the last
turn
      'input': {
         'text': ..., # raw user input as transcribed by the asr
         'hypotheses': [...] # tokenized asr confidence scores
      },
   'turn_no': ...,
   'nlu': {
      'annotations': {
         'intents': {...},
         'ner': {...},
         'processed_text': {...},
         'profanity': {...},
         'postag': {...}, # parts of speech tags
         'sentiment': {...},
         'topics': {...},
         ...
      },
      'modules':[...], # list of the nlu modules that run in the
pipeline
      'processed_text': {...} # user utterance after nlu
pre-processing has been performed
   },
   'previous_topics': [...], # list of all unique previous topics in
the session
   'response': {<bot_name>: <response>},
   'bot_states': {<bot_name>: {bot_attributes: {...},
                  # This is the dumping attribute, where each bot
can put attributes that needs to be stored in the db for whatever
reason (i.e. news_ids, into_bot flags, etc)
                  lock_requested: bool}},
}
```

FIGURE 5.3: The State Object's dictionary representation

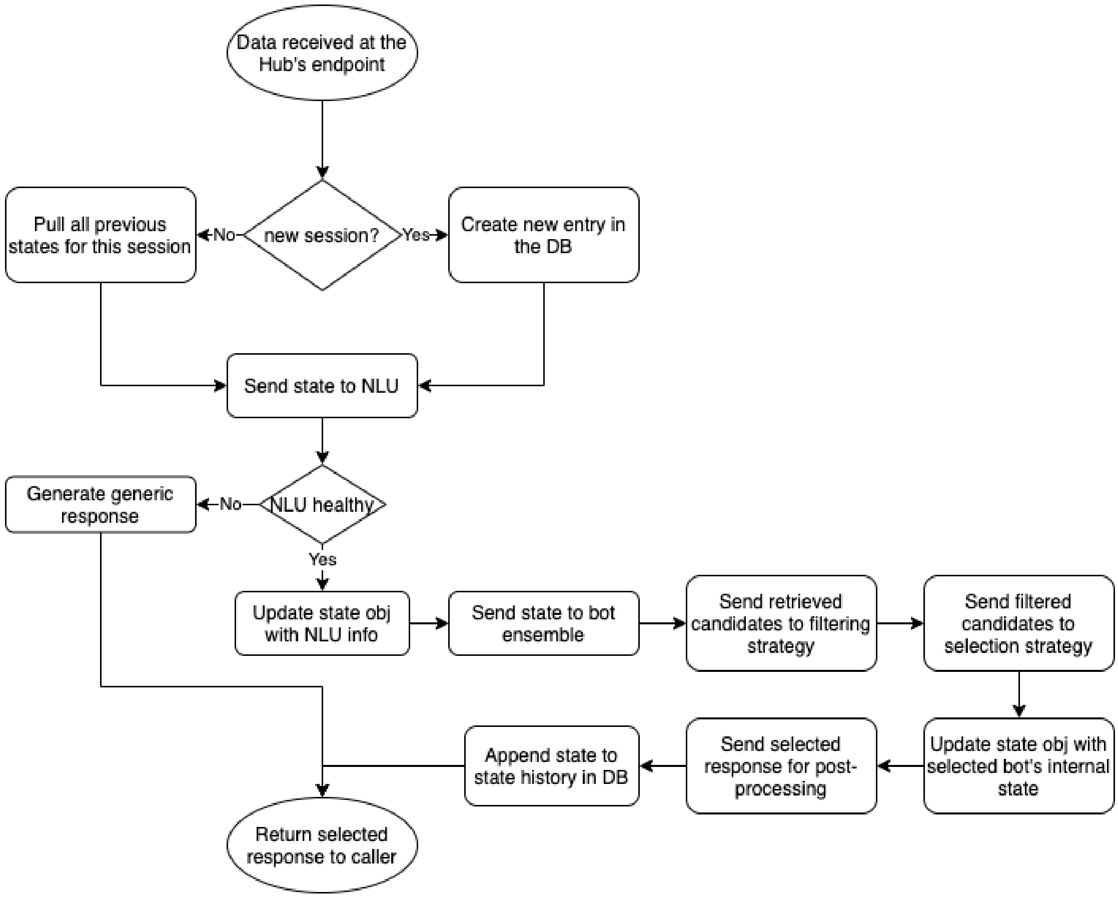


FIGURE 5.4: The 2018 Hub's pipeline. Hub acts mainly as the central message delivery component between the rest of the modules.

any internal resources they are using (e.g. pointers, pre-selected responses, etc) intact and available to be reused on subsequent turns in the conversation.

5.2 System Components

Several individual components of the system underwent an overhaul in order to meet the goals set forth by the team in 2018. Below follows a detailed description of these changes.

5.2.1 The Hub

One of the drawbacks of the monolithic architecture followed in 2017 was that the Bucket was overcrowded with features, bloating that component and making it susceptible to crashes due to load. Additionally, such practice increased the complexity of making changes to individual features in that pipeline as the competition was progressing. To that end, it was decided to completely redesign that central component and to de-couple it from its individual features. That new central component, called the *Hub*, had a single function which was to connect all the different modules in the architecture together and to pass information from one to another.

One of the most important changes to the architecture is that in 2018 only the Hub has access to the database (instead of each individual bot being able to store and retrieve its own data individually), minimising both the risk of data corruption and de-synchronisation of the conversation information. On each conversational turn, all of the conversation history up to that point and the user profile (as explained in section 5.2.4) are pulled from the database and are available to be accessed by the rest of the modules.

The Hub's operation is shown in Figure 5.4. Each turn, the Hub generates a new state object (Section 5.1.1) with the information received, and retrieves all previous states corresponding to the specific session received by the Lambda function (which composites the conversation history) from the database. The state object is then forwarded to the NLU pipeline, which further updates the state object with annotations, such as user's intents, named entities, POS tags, user sentiment, etc, as shown in Figure 5.5. In the case where the NLU module was not able to generate annotations (e.g. due to technical crashes or other malfunctions), a generic deflecting response was returned to the user instead (e.g. *"I'm sorry I didn't quite get that. Could you repeat that please?"*) in order to give the system the chance to recover (if able).

```
'annotations': {
   'intents': {...},
   'ner': {...},
   'processed_text': {...},
   'profanity': {...},
   'postag': {...},
   'sentiment': {...},
   'topics': {...},
   ...
   }
```

FIGURE 5.5: Alana's 2018 NLU annotations

The updated state object is then forwarded to the ensemble of bots simultaneously in a multi-threaded way, where each bot generates one or more candidate responses, which are then collected by the Hub. Each bot runs as a micro-service on a dedicated port, operates independently, and is agnostic to the operations or existence of the other bots in the ensemble.

Once all candidate responses are collected, they are initially filtered similarly to 2017 as described in Section 3.3.4. Then, a *selection strategy* is applied in order for the contextually best response to be selected. The selection strategy (Section 3.3.4) is defined by a Bot Priority List, which states which bots should handle the current turn, relying on probabilistic decisions with hand-crafted weights. In 2018, the Priority List followed the following order: *ClarificationBot* → *ProfanityBot* → *FactBot* → *WeatherBot* → *PersonaBot* → *[NewsBot, RedditBot, OntologyBot] (sharing the same priority)* → *WikiBot* → *EviBot* → *CoherenceBot*.

Once a response is selected, it is then post-processed (Section 5.2.5), by occasionally injecting the user's name (if known and appropriate) in grammatically appropriate places in the response, as well as adding conversational "drivers", in order to maintain the flow of the dialogue, as will be further described in the following sections. This ensures consistency and a more natural flow of conversation, despite the architecture followed not including a dedicated or centralised NLG component (as e.g. presented in alternative architectures in Chapter 2), but instead relying on each bot generating different candidate responses.

5.2.2 Natural Contextual Language Understanding

Another fundamental change in the 2018 architecture was the consolidation of all language understanding tasks (that were part of the Hub's pipeline in 2017) into a separate NLU module called *mercury* (named after the Roman god served as messenger). This module would accept the user's utterance along with all the information from the state object and provide annotations for that current utterance. This module is itself composed of a number of different modules $\mathcal{M} = \{M_1, M_2, \ldots, M_n\}$ each responsible for providing different types of annotations. A set of these modules $\mathcal{M}$ creates a pipeline P which is applied on the user's utterance, with each module being applied sequentially. This enables Alana to be more contextually aware by taking into account the different information present in the conversation such as topics, mentioned entities, and intents associated with the current utterance (Curry et al., 2018).

The variety of NEs used in the 2017 competition by both the user and the system motivated the team to seek ways to exploit the potential relationships between those entities to improve upon the system's engagement and coherence capabilities and the overall flow of conversation. To that end, the technique *Entity Linking* (Shen et al., 2015) was employed which utilises the annotation of various entities present in the Wikidata database with the linking capabilities of Yahoo's *Fast Entity Linking* (FEL) system (Blanco et al., 2015) to represent the links between those entities. This functionality would allow Alana to retrieve information on various entities that are relevant to each other, thus staying on topic longer and being more coherent in its responses.

The library of modules includes a variety of different annotation tools such as:

1. **Truecaser**: Based on a pretrained language model [1] this module autocapitalises the user's utterance, which in turn boosts the performance of the Named Entity Recognition (NER) module.

[1] https://github.com/nreimers/truecaser

2. **Contextual preprocessor**: a custom module which transforms the user's utterance using contextual information from the dialogue's history, such as ellipsis resolution (Table 5.1, turn 5) and indirect to direct questions (e.g. *"I don't know what X is"* into *"What is X?"*), as mentioned in Section 5.1. This transformation enables the information retrieval bots in the ensemble (e.g. OntologyBot) to respond to these queries. Other transformations include specific elliptical resolution on NEs when the system asked about the user's favourite entity (e.g. SYS: *"Who is your favourite singer?"* USR: "Dio" into *"my favorite singer is Dio"*, which enables better recognition by the user modelling mechanism as described later).

3. **POS tagger**: module developed around the *MorphoDiTa* Part-of-Speech tagger (Straková et al., 2014) which enables Mercury-NLU to annotate the user's utterance with POS tags.

4. **Regex-based Intent Recogniser**: Similar to 2017 regular expressions-based module which includes a compilation of regex patterns that are matched against the user's utterance to identify the intent.

5. **Neural Multi-Intent Classifier**: bi-directional LSTM classifier trained using real user data collected during the 2018 competition and manually annotated by the team. The classifier was able to recognise multiple intents in a single user utterance (e.g. *"I don't want to talk about politics anymore. Do you have any news about the Oscars?"* $\rightarrow$ intents: [dont_tell_about, news]). The classifier was trained on 2,000 user utterances from the collected interactions.

6. **Named Entity Recogniser Ensemble**: an ensemble of NER models which includes *SPaCy* NER [2] and *Stanford NER* (e.g. "*Is there any news about [North Korea]* (COUNTRY)?").

[2] https://spacy.io/models/en#en_core_web_lg

7. **Sentiment Analyser**: this module provides sentiment analysis annotations using the NLTK library (e.g. "*VADER is smart, handsome, and funny!*" [compound: 0.8439, neg: 0.0, neu: 0.248, pos: 0.752]).[3] [4]

8. **Entity Linker**: a module based on the *Fast Entity Linking* (FEL) system, which links entities present in the user's utterance and system's responses to entities in the Wikidata[5] database. These related entities are then used in downstream tasks performed by different bots, such as the *OntologyBot, ClarificationBot* or the *NewsBot*. The Entity Linker in this version of the system is able to provide identifiers to related entities under these categories: movies, books, music, video games, and sports.

9. **Entity Topic Classifier**: using the Entity Linker described above, the topic of the conversation can be determined by exploiting the properties that those entities are associated with in the knowledge base.

10. **Anaphora Resolution**: when a reference to an entity is detected in the user's utterance using a pronoun the annotations from both the NER ensemble and the Entity Linker are employed to associate it to the last mentioned (by either the user or the system) entity in the dialogue history (Table 5.1, turn 7). The Entity Linker's annotations are utilised to provide additional information to resolve to the correct gender or type of entity.

An example of the annotation provided by Mercury-NLU can be found in Appendix C.1

5.2.3 Bot Ensemble

Similarly to 2017, the bots were categorised into three types: rule-based, information-retrieval, and miscellaneous. Some of the bots (e.g. PersonaBot, ElizaBot) remained the same in terms of functionality, but they were updated to reflect more

[3] https://www.nltk.org/_modules/nltk/sentiment/vader.html
[4] https://www.nltk.org/howto/sentiment.html
[5] random example item from the Wikidata database https://www.wikidata.org/wiki/Q48924339

```
{
'current_state': {...}, # state object as described above
'history': [...], # a list of state dictionaries as above. Can be
   configured to only get the N last items
'user_attributes': {
      'user_id': ...,
      'user_name': "john",
      'map_attributes': {...}, # dictionary or various user attributes
   like list of likes and dislikes
      'last_session': ... # last sessionID the user engaged with
   }
}
```

FIGURE 5.6: The input structure of each bot in the ensemble

recent topics, and general language fixing in their outputs. In this section, the newly added bots and those that underwent considerable change are described.

5.2.3.1 Generic Bot Architecture

As described in Section 5.1, each bot in the ensemble runs in parallel and is called simultaneously. All the bots receive the exact same information (encoded in the State Object) as shown in figure 5.6, which includes the current state object (*current_state*), the previous n turns' state objects (*history*), and the *user_attributes* as described in section 5.2.4.

Having information shared across the bot ensemble alleviates the need for some bots to have to reprocess the user's utterance themselves (e.g. to extract NEs from the utterance) as they did in the 2017 version or to reload potentially heavy resources in memory that have been already loaded by other modules.

Given how the bots vastly differed in functionality, the only restrictions imposed on their implementation were to return their response(s) in a specific format, as shown in figure 5.7. Apart from its name identification (*bot_name*), each would produce from 0 to multiple candidate responses (*result*). In cases where the bot was in the middle of a transaction that would require multiple turns to be completed, a special flag could be used in order for the bot to "request" that it wished to handle

```
{
   'result': [...], # a list of possible response candidates
   'bot_name': ...,
   'lock_requested': bool, # Flag to state that a bot is requesting
 to handle the next turn as well. Multi-turn feature.
   'bot_params': {...} # the "helper attributes" that the bot
 requests to be saved. Multi-turn feature.
}
```

FIGURE 5.7: The output structure that each bot in the ensemble needs to return

this turn (*lock_requested*). Also, as explained in section 5.1.1, each bot was able to save its internal state in the state object by returning it as part of its response in the *bot_params* attribute of the output message.

For example, during turn 1, CoherenceBot might have asked the user *"Do you prefer sci-fi or fantasy?"* in which case it is expecting the user's response during the next turn. To keep track of that fact, it would have saved an internal flag in the *bot_params* stating that it is waiting for a specific user input. During the next turn, when it would receive the new state object, it would first look into its own parameters in the previous turn's state object (*history*) and check to see whether the user's current utterance resolves the question raised during the previous turn. Then, depending on the outcome, CoherenceBot would request from the selection strategy that it should handle this turn by utilising the *lock_requested* attribute.

The *lock_requested* attribute in combination with the *bot_params* is the mechanism that enables any bot in the ensemble to engage in multi-turn interaction with the users, since this allows the bot to focus and keep track of specific information relevant to that bot which can be delivered in multiple sections (e.g. in the snippet dialogue in Section 5.2.3.2).

5.2.3.2 WikiBot

WikiBot was redesigned in 2018 to utilise the changes in the architecture, especially in terms of system engagement in contextually-aware *multi-turn* conversation. Following the initial idea presented in Section 3.3.3, WikiBot uses an indexed Wikipedia dump, this time using MongoDB instead of Lucene, keeping the title and first sentence of each article. Attributes from the user's utterance (using the Entity Linker's annotations) are then used as search terms for querying the database. These attributes are (a) NEs (already recognised by Mercury-NLU), (b) NPs in the user's query, and (c) 1,2,3-ngrams (after being lemmatised and stop words being removed) of the user's utterance. The most relevant document is then retrieved and delivered to the user in sections (as shown in the example below). This information chunking was considered to enhance the auditory information retention of the users from a given article, given the interaction interface being a smart speaker.

Example responses

U: I liked John F. Kennedy

S: *(WikiBot)* Ok, I think you mentioned John F. Kennedy, I read that he served at the height of the Cold War, and the majority of his presidency dealt with managing relations with the Soviet Union... Let's chat more about John F. Kennedy.

U: sure

S: *(WikiBot)* Glad you're interested in John F. Kennedy. So, A member of the Democratic Party, Kennedy represented Massachusetts in the U.S. House of Representatives and Senate prior to becoming president. Shall I say more?

U: no

S: *(WikiBot + driver)* Sure, let's stop talking about John F. Kennedy. Who do you think made a big difference to American politics? For me it was Rosa Parks and Lee Harvey Oswald.

5.2.3.3 NewsBot

Building on top of the implementation of NewsBot described in Section 3.3.3, NewsBot in 2018 scrapes and indexes a number of reputable news sources (using the NewsAPI as before), but the articles can now be delivered in a multi-turn fashion as in WikiBot. Furthermore, NewsBot utilises the Entity Linker's annotations on the user's query, enabling this bot to retrieve *related* news articles to entities the user mentioned. This is meant to provide a more natural and coherent flow across turns.

Example responses

U: Tell me the news

S: *(NewsBot)* `*username*`, I saw this news on The Wall Street Journal with the headline, How Washington Learned to Love the Deficit. Do you want to learn more about this ?

U: No

S: *(NewsBot)* This might not be directly related, but `*username*`, I have another interesting piece of news from Reuters with the headline, U.S. retailers' halting outlook reveals scale of tariff fear. Does it sound interesting to you?

U: Yes!

In the example above, the intent classifier recognised that the user did not wish to continue in the second user turn, in which case the NewsBot (using the Linked Entities provided by the NLU module about that entity) attempts to re-engage the user in the conversation by offering a related article in that topic (politics). The user at any point is free to switch topic or continue with the current conversation.

The `*username*` tag used used in this example is further explained in Section 5.2.3.8.

5.2.3.4 OntologyBot

In an open-domain conversation the system should be able to suggest interesting topics to avoid the conversation getting stuck and to keep a natural flow of topic shifting throughout the conversation. This bot makes use of the linked entities provided by the Entity Linker NLU module, where every time a NE is mentioned by the user, Alana tries to discover connections in the Wikidata graph that relate that entity with other domain related entities in the knowledge base.

The OntologyBot employs SPARQL queries[6] to link a source entity present in the user's utterance to a related final one through a multi-hop process, in the same domain/category (movies, music, books, video games, or sports). The source and final entities are then wrapped in pre-defined NLG templates, along with a few intermittent entities in the linking chain to generate a final utterance that will hopefully drive the conversation further.

Example responses

U: Actually, my favourite movie is Blade Runner.

S: *(OntologyBot)* Oh I really like *Blade Runner*. I guess you know that one of the lead roles in *Blade Runner* was played by *Harrison Ford*. *Harrison Ford* also starred in *Star Wars The Force Awakens*. So, what's another movie like *Blade Runner* that you've enjoyed?

As shown in the example dialogue and in Figure 5.8, using a detected source entity in the user's utterance (Blade Runner), OntologyBot is using the linked entities provided by the NLU to extract a related entity at random (Detective Decker), which is in turn linked to a third entity (Harrison Ford). This is then linked to another entity (Han Solo) which leads to the final hop (Star Wars). These entities are then used as slots in a template to generate the final system response in the example.

[6] https://www.w3.org/TR/rdf-sparql-query/

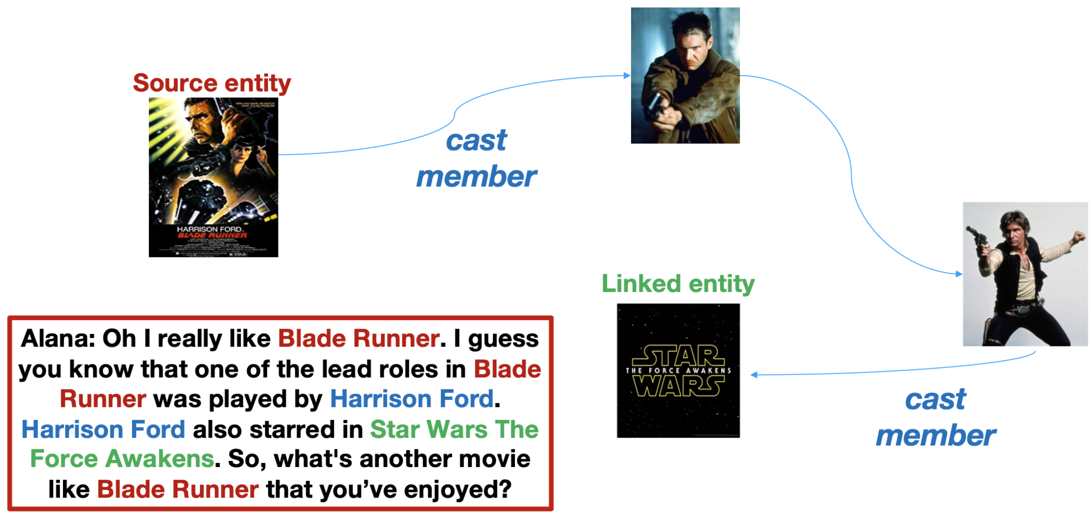


FIGURE 5.8: Entity Linking example using the FEL system to link entities in the user's utterance with related entities in the Wikidata graph. Image provided by Alessandro Suglia.

5.2.3.5 ClarificationBot

With the addition of Entity Linking in 2018, there are a number of occasions where the user mentions an ambiguous entity which falls under multiple topics and a topic is not already specified (e.g. *"Tell me about Blade Runner"* $\rightarrow$ could refer to the 1982 Blade Runner movie, the Blade Runner 2049, or the South African Paralympic athlete). In this cases, it was decided to clarify this information with the user, making sure the final response provided by the bots is grounded on the correct topic. This was realised by utilising the Entity Linker annotations to produce clarification questions (e.g. `''Do you mean Blade Runner the sci-fi movie or the South African Paralympic athlete?''`; see also Table 5.1, turn 13). The user then would be able to clarify which option they were referring too, in which case the Entity Linker's annotations would be updated to reflect on the correct topic on the following turn.

Example responses

U: Let's talk about Blade Runner

S: *(ClarificationBot)* I'm sorry. Do you mean Blade Runner the sci-fi movie or the South African Paralympic athlete?

U: The movie

5.2.3.6 RedditBot

Sometimes generic topics like "pizza", "sleeping", etc might not have an associated Wiki or News related information to be retrieved. This is quite common in open-domain conversations which leads to more general topical chat. To that end, another information retrieval bot based on the open-source framework Lucene[7] was added, which scrapes, indexes, and searches popular subreddits like "ShowerThoughts" and "Today I Learned". This bot searches the index for relevant Linked Entities, NEs, and NPs (in that order) mentioned in the user utterance.

Example responses

U: I like pizza

S: *(RedditBot)* Whenever a pizza commercial has stretchy melty cheese, it looks good and appealing, but when it happens in real life it's annoying and messy.

U: I like sharks

S: *(RedditBot)* Martin luther king jr. only gets a day, but sharks get a week.

U: Tell me about cats

S: *(RedditBot)* The only time cats display true happiness on their face is when they are sleeping.

[7] https://lucene.apache.org

5.2.3.7 Abuse Mitigation Bot

By analysing the data collected in the 2017 competition, it was found that many interactions contained some form of profanity or abuse towards the system. To mitigate this behaviour, an abuse-detection model was trained on 4,050 manually annotated examples from the Alana 2017 dataset, followed by specialised designed response strategies. The embedding-based abuse detection model is able to distinguish between non-abusive, generally offensive, and sexually-charged utterances or hate-speech. This functionality however does not focus on mitigating user foul language in general, but foul language specifically targeted at the system.

Example responses

U: F*** you!

S: *(ProfanityBot)* We can be pretty unicorns together! What a great idea you had!.

U: Cool!

S: *(ProfanityBot)* It's all good. *username*, I was wondering, who is a rock star you would like to meet? I want to meet Kanye West.

The response strategies designed to mitigate such behaviour include changing topic, appealing to authority (e.g. *"Would you like me to send a transcript of this conversation to your contact: Mum?"*), chastising the user (e.g. *"Do you talk to your mother like that?"*), etc. These strategies and further discussion on the motivation of this component are outlined in Curry and Rieser (2019).

5.2.3.8 Coherence Bot

Although during the 2017 competition the *IntroBot, RapportBot* and *Coherence-Bot* were integral to maintaining a sense of coherence in the conversation, their functionality was heavily convoluted and extremely specific. Furthermore, heavy refactoring was needed to their functionalities in order to follow the changes in the system's logic in 2018, as described earlier. Finally, given the close interaction and goals between the three bots, it was decided that they can be consolidated into a single-scope *CoherenceBot*. This also enabled the optimisation of the coherence strategy as will be explain in section 5.3.

CoherenceBot's main functionality is to keep the conversation flowing in an engaging and coherent way, in the case that none of the other bots manages to produce a response, and thereby acting as the *fall-back strategy* of the system. The CoherenceBot works by initially building a simple user model based on the user's preferences gathered during the initial phase of the conversation and then gradually updating it throughout the conversation. For example, a user may like sci-fi and books, but dislikes politics. This user model can be used later when trying to perform a topic shift. In 2018 we wanted the system to be restricted to only be responsive to user requests, but engage in the conversation as an equal participant. The CoherenceBot is the only bot in the ensemble that utilises this *mixed-initiative* capability (Buck et al., 2018), being able to switch the topic if the conversation demands it. One of the core concepts of the CoherenceBot is the usage of conversational "drivers". These are questions or statements on a topic that try to drive the conversation further, to avoid reaching a conversational dead-end. These drivers can either be output on their own, or appended to other bots' responses that would otherwise probably create a conversational stop (e.g. User: *"Do you like movies?"* System: *"Yes"*, does not drive the conversation forward as much as *System: "Yes. Which is your favourite movie?"* would do). This functionality is further explained in Section 5.2.5.

These drivers are divided into 5 categories:

- **Topic**: Drivers on specific topics such as movies, music, etc (e.g. *"What was one of your favourite movies growing up? I loved the Lion King."*)
- **Preference**: These are the same as the topic drivers, but aligned to the user's individual preferences (see Sec.5.2.4)

- **Rapport**: These are predefined mini-scripts (2 turns depth), aiming to expand the user model by asking the user targeted questions. (e.g. *"Are you more interested in reading books or watching movies?"*)

- **Chit-Chat**: These drivers are aimed at engaging in small talk with the user, but also to elicit a NE from the user to drive the conversation further, using the other bots in the ensemble. However, these drivers were designed in such a way that even if the user's follow-up utterance didn't provide any information to continue the conversation, the system would still manage to follow-up on it's own response. (e.g. *SYS: "What's a song you love to rock out to?" USR: <incomprehensible utterance or no-intent detected> SYS: "I love Whatever It Takes, by Imagine Dragons."*)

- **Advertising**: Drivers specifically designed to inform the user of some of the system's capabilities (e.g. *"Don't forget you can talk to me about lots of things, like movies or music or the news. Also I can sing for you."*)

- **Generic**: These drivers are aimed at engaging the user in conversational topics the system can handle (e.g. *"I was wondering. Do you prefer talking about {pref1} or {pref2}?"*). The slots ({*pref1*}, {*pref2*},...) are dynamically filled from the user model, trying to engage in topics that the user previously showed interest in.

Rapport category drivers follow a constrained path, where once a question has been asked to the user, there is only a certain set of valid responses the user can provide (usually a response to a binary question). If the question/driver isn't answered by the user during the next turn, that question is discarded from the available driver pool to avoid repetition. An example of this would be when the system asks: *"Do you prefer reading books or watching movies"* where the user is expected to respond with either of these options, both, neither of them (explicitly), or state a dislike towards any of them. Depending on whether the user expressed a preference or dislike the user model (section 5.2.4) is updated.

Another important feature of CoherenceBot is to handle cases where the user was using intermittent pauses in his speech, causing the ASR to start processing the user's utterance prematurely, leading to only partial (incomplete) sentences (e.g. SYS: *What is a movie that you like?* USR: *"I like... <long pause> Jurassic Park (which was not captured by the speech recogniser due to the long preceding pause)"*). This was detected by Alana's NLU intent classifier as an *incomplete_sentence* intent and handled by CoherenceBot by providing specifically designed responses to help the user in subsequent turns (e.g. *"That's fine. I know it's not an easy question. Take your time."*).

As stated, every driver available in the CoherenceBot is able to either be output over a single or multiple turns. This is done by separating each segment of the driver with a special symbol, such as in the driver: *"What's a song you love to rock out to? ∼∼ I love Whatever It Takes, by Imagine Dragons."*. The reasoning behind this is to give the user a chance to get back into a specific topic on their own (e.g. if the topic is *music*), the bot indirectly prompts the user for a band/singer/etc, otherwise it provides one itself during the next turn (the next segment).

Since the *CoherenceBot* also acts as the system's fallback strategy, it is crucial that it returns a candidate response in the list of candidates at every turn. The response is in the form of a conversational driver, based on several features provided by the NLU module (Curry et al., 2018). Each time the bot is called, it looks into features from the state object such as *topic, turn number, intent, user preferences, user dislikes*, as well as its own internal state (*staged driver/segment, previous topic discussed*, etc), in order to decide which type of driver it should pick next to drive the conversation further.

As shown in Fig. 5.9, the bot initially takes initiative by asking for the user's name as well as a few binary questions, in order to steer the conversation later towards topics that the user is interested in (as previously described in section 3.3.3.10). In the case that in the previous turn a rapport type question was asked, it checks whether the user response satisfies the staged question. If on

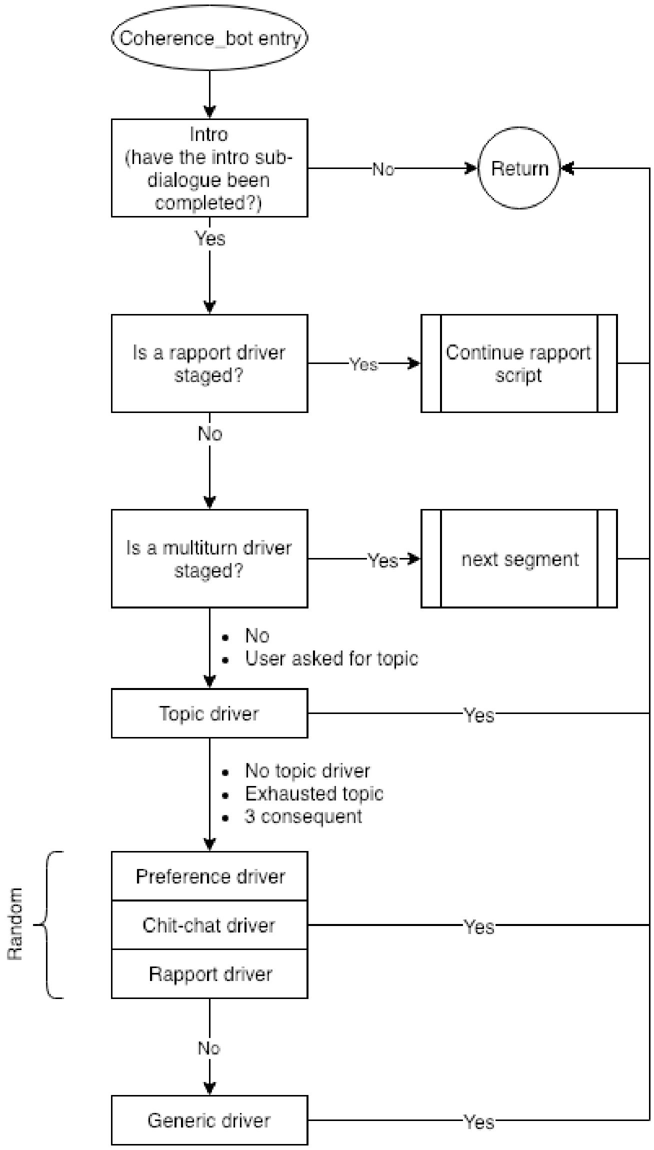


FIGURE 5.9: The CoherenceBot initial driver selection strategy randomises the selection of driver categories in the bracket.

the other hand part of a multi-turn driver was returned during the last turn, the next segment is returned, unless the user explicitly requests a specific topic. If none of these conditions are encountered, the new driver is selected in the order of *topic* →[*preference, chit-chat, rapport*] (with *preference, chit-chat* and *rapport* sharing the same probability) →*generic*. *Advertising* drivers were offered instead of a *generic* one with a probability of 0.3.

Further conditions were put into place to handle cases where sequential user utterances did not provide any useful information (e.g. due to low ASR confidence score or continuous non-existent intent recognised), possibly leading to CoherenceBot getting "stuck" by repeatedly providing drivers from the same category. To address this, *topical* and *rapport* drivers would be provided up to 3 subsequent times.

By default, all drivers apart from the *generic* ones can only be used once per dialogue, meaning that in longer dialogues, for example a *topic* driver may be required, but due to having no more drivers available on that topic, the bot will progress on to the next step (as the CoherenceBot must *always* return a response).

An example of interaction with CoherenceBot is presented in Section 5.3, where optimisation of the strategy shown in Figure 5.9 is further explored.

5.2.4 User Modelling

Throughout the conversation, the CoherenceBot utilises and updates the user model, in order to (a) steer the conversation towards paths that interest the user (Table 5.1, turn 11) and (b), identify returning users. This is done by:

- Learning, storing, and using the user's name throughout the interaction.
- Asking the user a specific question (using the *rapport* drivers).
- Using the NLU intent schema that identifies explicit user preferences (e.g. *"I really like X"* where $X \in T\{topics\}$).

This persistent model is represented in the database in the following format, and is kept across sessions the same user is engaged in:

```
{
  "map_attributes": {
    "dislikes": [
    "cats",
    "movies"
```

```
        ],
        "preferences": [
        "sports",
        "movies",
        "basketball"
        ]
    },
    "user_id": "xxxx"
}
```


The user is identified during the first few turns of the conversation (as explained in Section 3.3.3.9). With the changes in CoherenceBot's functionality and the addition of the State Object (Section 5.1.1), however, returning users can be re-identified using the user's Amazon Alexa device's ID (which is provided by the lambda function - see Section 5.1) allowing the re-usability of a previously established user model for that user. Given however the fact that in 2018, Amazon hadn't released user identification on their devices, the system had no automatic way of distinguishing between users in the same household sharing a device. In these cases, clarification was required before the system was able to identify them as a returning user.

Example use case of a returning user:

S: *(CoherenceBot)* Hi. This is Alex I am talking to, right? *(Alana found the same device ID - which is used as a user identification)*

U: Yes, that's right.

S: *(CoherenceBot)* Welcome back then, Alex! So, since I know you like movies, what is a movie that you watched recently?

Example use case of a user using a shared device:

S: *(CoherenceBot)* Hi. If I am not mistaken, this is Alex, right? *(Alana found the same device ID)*

U: No

S: *(CoherenceBot)* So. Could you tell me your name? *(new user profile is created)*

U: It is John.

S: *(CoherenceBot)* Pleasure talking with you John. Let's get to know one another a bit better.

This model is crucial to the functionality of CoherenceBot as it needs to suggest topics that would increase the user's engagement towards the system and not topics that the user already stated his aversion towards.

5.2.5 Post Processing

As depicted in Figure 5.4, the last step before the selected response is sent back to the user is post-processing. This process builds on top of the 2017 one described in Section 3.3.4. `*username*` and `*driver*` special tags can freely be used by any bot in the ensemble if their internal functionality requires it (e.g. NewsBot might use a driver as a deflection strategy in case there is no more information to be offered to the user. However, given the parallel and concurrent execution of the bots in the ensemble, CoherenceBot can not be forcefully called in this architecture, but instead the Post-Processor joins the responses from the requested bot and the output driver already generated by CoherenceBot that is present in the pool of candidates (since CoherenceBot *always* produces a response). This change, alongside the ability of the CoherenceBot to respond across multiple turns, can imitate very natural deflection processes present in human-human communication like in the example dialogue below.

Post-processing example

S: *(CoherenceBot)* What do you think is a good movie about technology?

U: What's your favorite thing to learn?

S: *(ElizaBot)* I love to spend my free time reading about artificial intelligence. *(CoherenceBot)* Anyway, I believe a great movie about technology was the Social Network as well as the Imitation Game.

5.3 Evaluation of coherence bot strategies

Given the various challenges in open-domain conversation presented in Chapter 1, quite often the quality of a human-agent interaction can degrade due to conversational dead-ends (i.e. when a participant has nothing more to say on a topic), poor language understanding (for instance due to noisy environments), or failed topic switching. Although CoherenceBot was designed and implemented in 2018 specifically to mitigate these challenges, it is not always clear *which* action (out of the available ones as described in Section 5.2.3.8) should be taken at each stage of the conversation. It is worth reiterating that CoherenceBot holds the lowest priority in the Priority List (Section 5.2.1) when none of its locking triggers have been invoked (e.g. the user asks explicitly to talk about a certain topic such as movies or music). Also CoherenceBot has to provide a candidate response on every turn, so the system always has a fall-back respond to return to the user in cases none of the other bots provided a sensible answer or even if the user's utterance didn't provide any meaningful information. However, the actions CoherenceBot takes each turn have been hand-crafted and are semi-fixed in each session (as shown in Figure 5.9), which raises the question whether they can be further optimised. Additionally, CoherenceBot is not dependent on the raw user's utterance, but rather specific NLU annotations, making the action space to be explored finite. Thus, the optimisation can be casts as an MDP problem and RL can be employed to train a policy for CoherenceBot.

5.3.1 Experiment: Building the baseline

Initially, two baseline strategies were created by gathering 34,144 conversations with real users during the Alexa Prize contest in 2018 over a period of 29 days (As a reminder, at the end of the interaction, each user rated the overall conversation using a 5-point Likert scale).

In one of the baseline strategies the driver categories were picked at random uniformly (with the exception being when the user explicitly wants to talk about a specific topic), while the other used the handcrafted semi-stochastic (Handcrafted SS) driver selection strategy as described in Section 5.2.3.8.

	Hand-crafted SS (condition 1)	**Random Baseline (condition 2)**
Num dialogues	17,072	17,072
Num of turns	308,584	394,815
Coherence_bot responses	163,801 (53.08%)	165,284 (41.86%)
Avg num of turns	18.07	**18.16**
Avg rating	**3.549**	3.502

Table 5.2: Evaluation results for the two conditions

Table 5.2 shows that the Hand-crafted strategy improves only slightly over the Random strategy.

5.3.2 Experiment: Strategy Implementation

Following the creation of the baseline strategies, an RL environment setup was designed using the popular *Q-learning* technique (Sutton, 1998).

The state space S was defined by the attributes:

- topic $t \in T$, where T are the topics handled by CoherenceBot as shown in Appendix A,

- intent $i \in I$, where I denotes the different intents provided by the intent classifier,
- topic in dislikes $td \in [0, 1]$, if the user has already shown their disapproval for the current topic
- staged multi-turn driver $mt \in [0, 1]$, depending on whether a section of a multiturn driver has already been given in the previous turn,
- rapport staged $r \in [0, 1]$, depending on whether a rapport mini-script was initiated in the previous turn,
- bypass rapport $br \in [0, 1]$, depending on whether the limit of 3 subsequent rapport drivers have been provided,
- end episode $e \in [0, 1]$, denoting the end of a given dialogue

The action space A included 7 different actions CoherenceBot would be able to take: TOPIC, PREFERENCE, RAPPORT, GENERIC, MULTITURN, KEEP-GOING, and INTRO. These correspond to the different driver categories available as shown in Figure 5.9, including the option to provide a new multi-turn driver (MULTITURN) or the next segment of an already staged multi-turn driver (KEEP-GOING).

The learning rate r was set at 0.81, the epsilon e at 0.1 and the discount factor γ at 0.96 to force focus on the previous turn. The q-values were initialised with zeros (see Equation 2.3).

Using this environment, two additional policies were trained on the Alana 2018 data collected near the end of the competition. One policy optimised for longer engagement (providing additional reward for each consequent user turn) while the other optimised for user rating (rewarding proportionally to the final user rating given to the dialogue). Each policy was then deployed on the live 2018 Alana system for 3 days, after the end of the 2018 competition (while the Alana system was still online in North-America) deciding what action the CoherenceBot should take on each turn. The results are shown in Table 5.3.

	policy	**dialogues**	**user rating**
P0	hand-crafted	635	**3.695**
P1	random	607	3.681
P2	user rating optimised	691	3.606
P3	duration optimised	593	3.636

Table 5.3: Evaluation results of the 4 different policies. Each policy was evaluated on a similar number of dialogues.

The results show that the hand-crafted policy P_{hc} outperformed the rest, however they were all quite similar in terms of user ratings. Although there are multiple confounding variables in this A/B testing environment (e.g. ratings being affected by the responses of the other bots in the ensemble), but given the very high proportion of total turns in a dialogue CoherenceBot is handling, it shows that the initial strategy followed in the implementation of CoherenceBot provides good results, and that the policy at minimum learned how to imitate that.

5.4 Discussion

In this chapter, the updated architecture and functionality of the Alana system was described, as an alternative to the traditional Spoken Dialogue System architecture for open-domain conversation. This architecture uses an ensemble of different bots as the NLG component, competing for selection using a mixture of response ranking and a trained policy for topic selection (provided by the CoherenceBot). All updated as well as newly developed components were described, focusing on how they are able to enable multi-turn, open-domain dialogue with the users.

Lastly, a RL experiment conducted after the end of the 2018 competition was described, optimising the actions the CoherenceBot should take on each turn in terms of conversational driver selection. Two different policies were trained over the course of 3 days, alongside with a random policy acting as a baseline. Interestingly, it was found that the random policy scored slightly higher (3.681) over the two optimised policies, one optimising for user rating, while the other for dialogue duration, scoring 3.606 and 3.636 respectively. The 3-day evaluation showed that the hand-crafted policy was still performing slightly better than the rest (3.695). These results could be attributed once again to the noisy sparse user rating signal and the different number of confounding variables, which underlines the challenging task of selecting a suitable set of features for the reward function. Additionally, the 3-day evaluation period might have been too short, and ultimately further investigation of the policies is required to draw more informative conclusions.

In the next chapter, the data collected during the semi-finals and finals stage of the 2018 competition (the *AL2018* dataset) will be analysed.

Chapter 6

Alana 2018 Dataset Analysis

Similarly to Chapter 4, the aim of the work presented in this chapter is to explore possible relationships in the data collected during the Amazon Alexa Prize Challenge 2018 competition and identify which features can be further exploited to increase the overall quality of the conversation.

6.1 Dataset description

The data collected in the Alexa Prize Challenge 2018 (henceforth named **AL2018**) include all interactions from the beginning of the semi-finals period of the 2018 competition onward. For the purposes of comparison with the previous year's dataset (AL2017), a similar period of interactions was selected, from 01-07-2018 to 30-11-2018 (∼5 months worth of interactions). In 2018, additional ASR information was logged as presented in Table 6.1. The AL2018 consists of 92,722 dialogues (sessions) and 1,595,015 full turns[1]. This year the average length of the user's utterances was 3.89 words per user turn, similar to that of the previous year (3.79). The system's average utterance length however was significantly increased to 26.87 words per system turn (from 21.91).

[1]A full turn consists of a user and a system's turn

	rating	user utt confidence score	num of turns	utterance length (user)	utterance length (system)	vocabulary size (user)	vocabulary size (system)
mean	3.62	0.82	37.76	3.89	26.87	3.87	22.99
std	1.38	0.17	35.08	3.78	16.43	3.39	11.84
min	1.00	0.00	1.00	1.00	1.00	1.00	1.00
25%	3.00	0.76	15.00	1.00	14.00	1.00	14.00
50%	4.00	0.88	28.00	3.00	25.00	3.00	22.00
75%	5.00	0.94	49	5.00	35.00	5.00	30.00
max	5.00	1.00	364	83	293	65	201

Table 6.1: AL2018 aggregated statistics.

num of dialogues	92,722
num of turns	1,595,015
vocabulary size (user)	51,119
vocabulary size (system)	266,161
avg user utterance length	3.89
avg system utterance length	26.87

Table 6.2: AL2018 overall statistics

In Figure 6.2 a zoomed in version of the ratings in 2018 is presented. The sudden dip in ratings around 05-10-2018 is attributed to the sudden huge influx of sessions received during that period (Figure 6.3) which also increased the number of timeout errors in the system. Although there is a slight increase in the overall ratings near the end of the period, ratings seem to plateau around the 3.65 mark.

To confirm this plateau effect, and to put the average ratings into perspective, the winning systems of the 2019 and 2020 Amazon Alexa Prize competitions in which our system did not compete (*Emora*, from Emory University and *Alquist*, from Czech Technical University), received an average of 3.81 (with 2nd place at 3.17 and 3rd at 3.14)[2] and 3.28[3] respectively. This is also supported by Venkatesh et al. (2018b), where the overall average of all Alexa Prize social bots user ratings was 3, while the average of all Frequent Users (returning users) ratings was 2.8.

[2]https://www.amazon.science/latest-news/amazon-announces-2020-alexa-prize-winner-emory-university

[3]https://voicebot.ai/2021/08/16/alexa-prize-grand-challenge-4-awarded-to-team-from-czech-technical-university/

FIGURE 6.1: Average daily ratings during both 2017 and 2018 competitions. Please note the ~9-months gap between the two competitions.

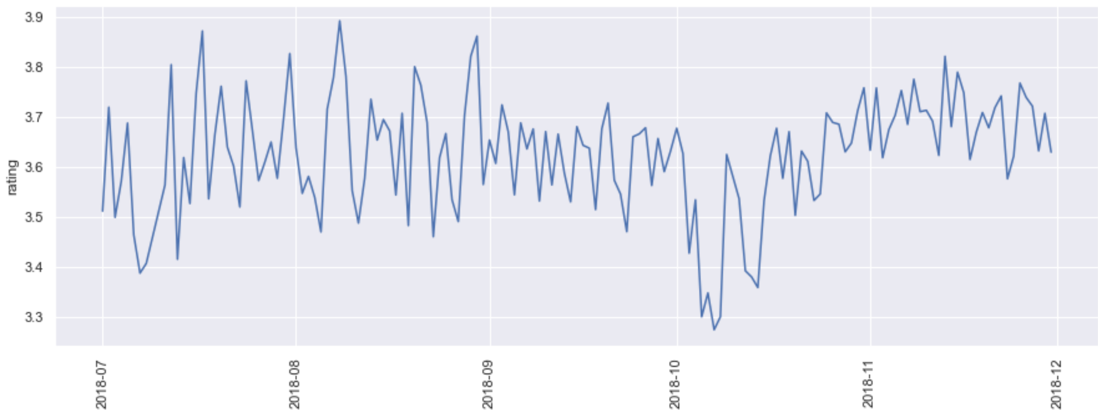


FIGURE 6.2: Alexa Prize Challenge 2018 ratings. The sudden jumps in ratings are attributed to small number of calls in these days. The last couple of months in the competition the system performed consistently well.

In Table 6.3 the updated bot usage distribution is shown, alongside the difference in percentage from the previous year's competition. As described in Section 5.2.3.8, *IntroBot*, *RapportBot* and *CoherenceBot* were integrated into a single *CoherenceBot* in 2018. Thus, the percentage difference calculation for coherence_bot used the combined percentages of those three bots in 2017 (total of 164,600 turns in 2017). The amount of functionality that was merged into that single bot in 2018 (including the "fallback" strategy role it was serving as described in Section 5.2.3.8), led CoherenceBot to handle almost half of all system responses. Yet, WikiBot showed

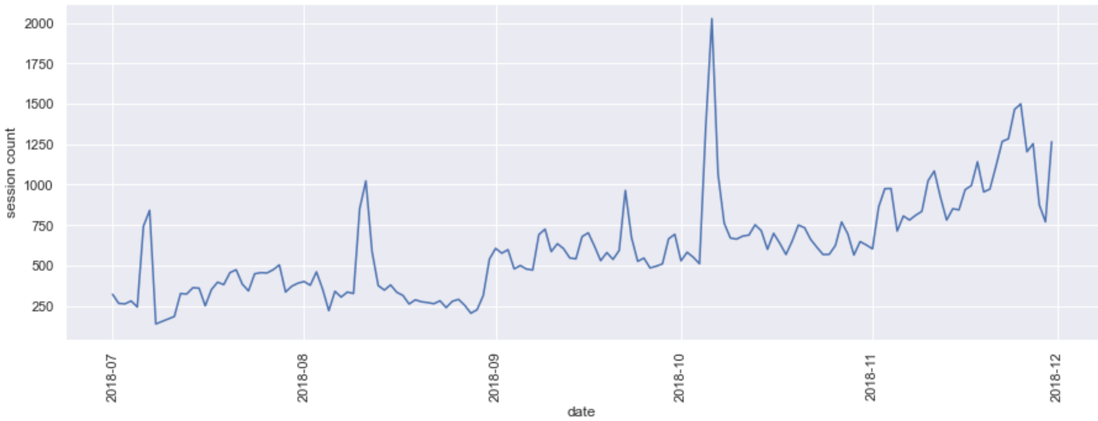


FIGURE 6.3: Number of sessions received per day. The spike in October aligns with the drop of ratings in Fig 6.2

	num of responses	**%**	**% difference from 2017**
coherence_bot	782587	49.06	+36.39
news_api	144459	9.05	-6.35
wiki_bot	128,537	8.05	+4.47
ontology_bot	115127	7.21	N/A
news_bot	111506	6.99	-8.41
persona_bot	94038	5.89	-12.14
eliza_bot	69416	4.32	-12.14
reddit_bot	65858	4.12	N/A
evi_bot	57455	3.60	-3.35
clarification_bot	18249	1.14	N/A
fact_bot	5863	0.36	-5.88
empty_hub	978	0.06	N/A
weather_bot	496	0.03	-0.04
profanity_bot	446	0.02	N/A

TABLE 6.3: Distribution of bot usage in 2018, including the difference in percentage from the 2017 competition. *Empty_bucket* is a special type of response generator fired in the rare occasion where no response was generated by any of the bots (e.g. due to complete system crash or other technical issues) and a generic response was returned instead.

a 4.47% increase in usage in 2018. This can be attributed to the multi-turn capabilities added to WikiBot in 2018.

6.2 Correlation between dialogue's length and rating

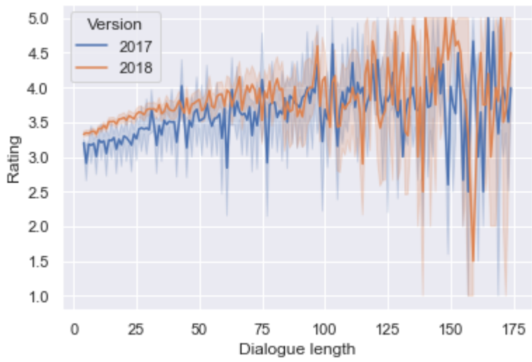


FIGURE 6.4: Relationship between rating and the average dialogue length (number of turns) per session for 2017 and 2018. The faded areas denote aggregation over multiple y values at each value of x and shows an estimate of the central tendency and a confidence interval for that estimate.

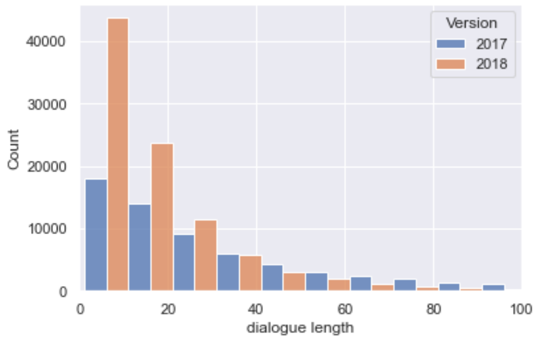


FIGURE 6.5: Distribution of dialogue length per session. Each bin holds the number of dialogues with number of turns in increments of ~ 10.

Following the same methodology and logic as in Chapter 4, the correlation between the dialogue's length and rating is investigated. Figure 6.5 shows that the AL2018 dataset follows a similar distribution shape to the 2017 counterpart.

The correlation found was: Spearman ($\rho = 0.1192$), Kendall ($\tau = 0.0898$) showing again a weak correlation between dialogue length and rating. Figure 6.4 shows the lineplot between the dependant variable (rating) in both years. In 2018 a more steady variation in ratings is observed as the dialogue's length increases, however it varies excessively on longer dialogues (over 100 turns).

6.3 Correlation between user/system utterance length and rating

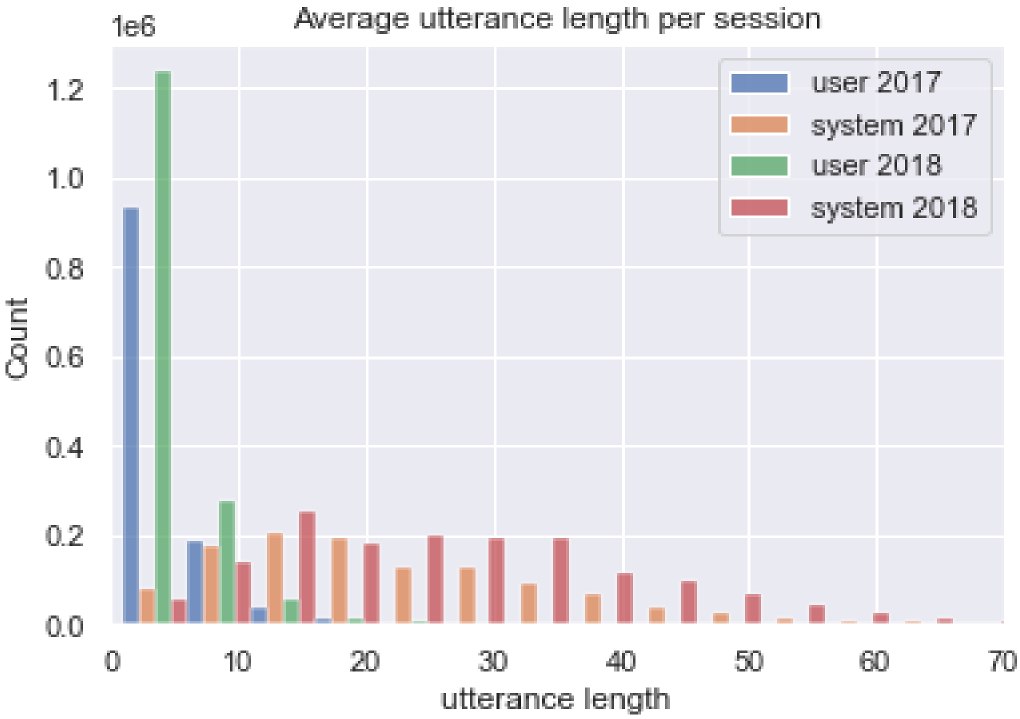


FIGURE 6.6: Average user and system utterance length per dialogue for both 2017 and 2018 versions of Alana

Similarly, the correlation between the utterance length for both the users and the system with the dialogue's rating is investigated. The findings are summarised in Table 6.4. The outliers were calculated at 25 and 80 (± 3.29 standard deviations from the mean) words long for the user and system utterances respectively. The similar shape of the distribution between the 2017 and 2018 systems shown in Figure 6.6 and Figures 6.7, 6.8 shows a consistent result between the versions.

	Spearman-ρ	Kendall-τ
User	0.0722	0.0535
System	0.0552	0.0404

TABLE 6.4: Correlations between user and system utterance length and dialogue ratings.

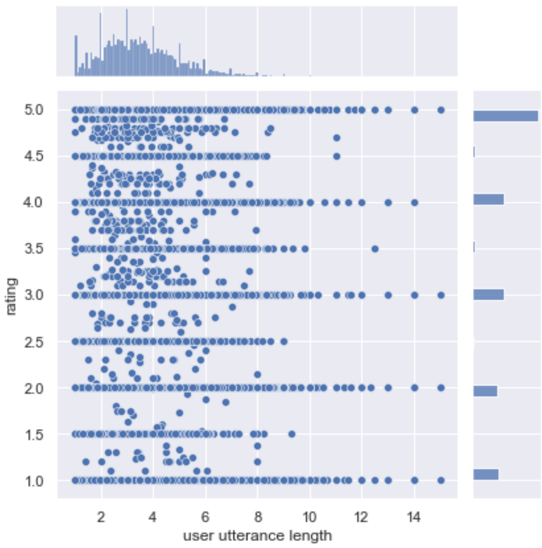


FIGURE 6.7: User avg turn length in 2018 scatter plot with additional distributions of ratings and length.

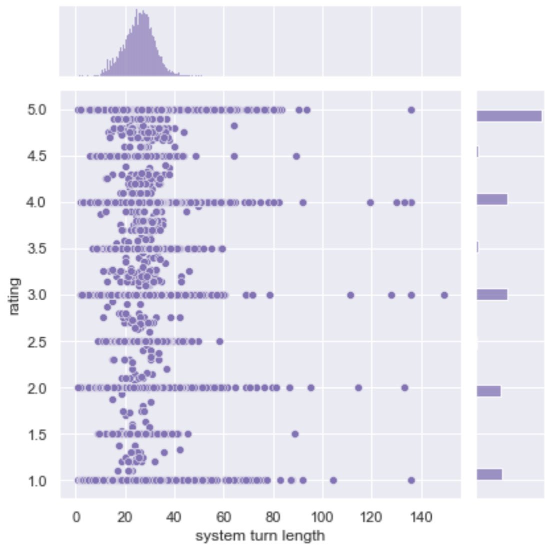
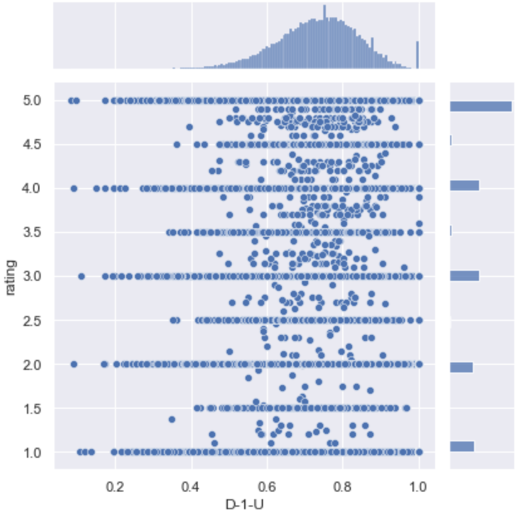


FIGURE 6.8: System avg turn length scatter plot in 2018 with additional distributions of ratings and length.

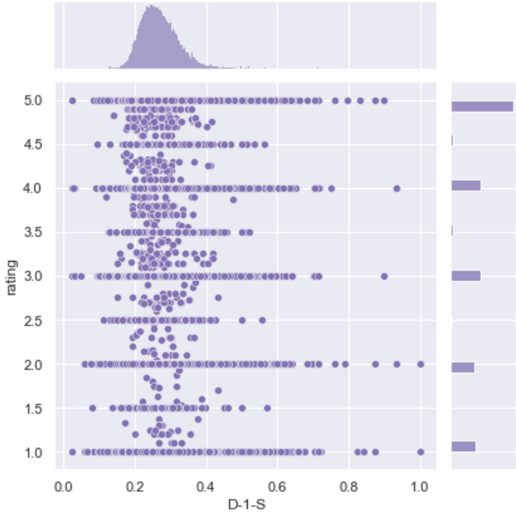
6.4 Correlation between lexical diversity and rating

Similarly to Section 4.4, again the unique uni- and bi-grams (D-1 and D-2 respectively) were calculated on the summed pool of all user and all system utterances, which are summarised in Table 6.5 and Figure 6.9. Furthermore, Figure 6.10 shows the average D-1 and D-2 for both users and the system utterances per session for both versions of the system.

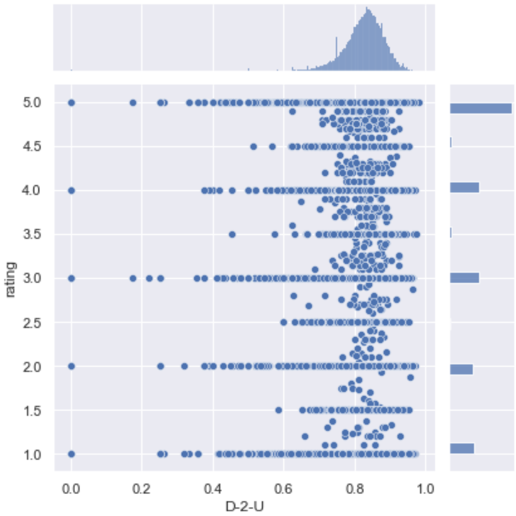
No significant change was observed in the 2018 dataset compared to AL2017 with regards to correlation between lexical diversity and conversation rating.



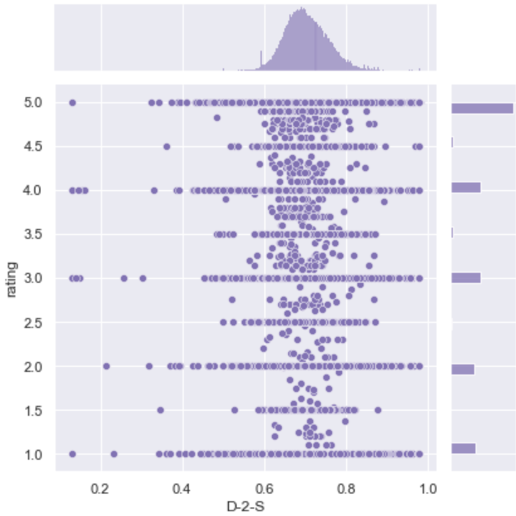
(A) Distinct uni-gram across all *user* utterances



(B) Distinct uni-gram across all *system* utterances



(C) Distinct bi-gram across all *user* utterances

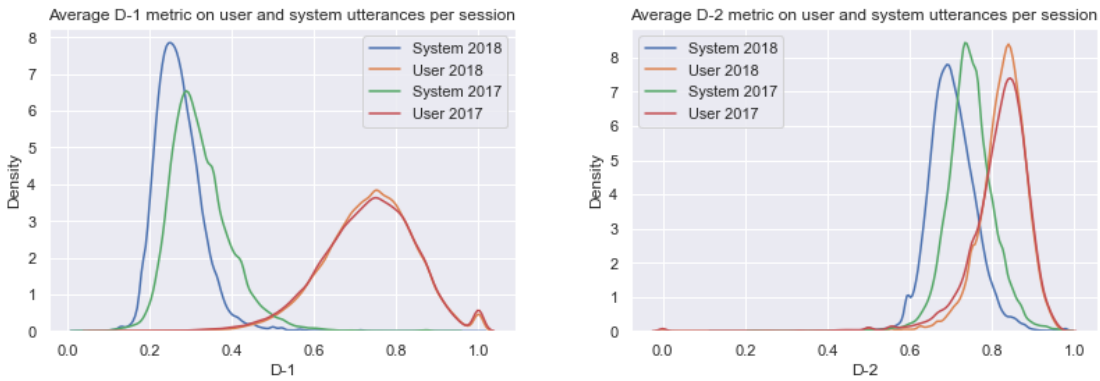


(D) Distinct bi-gram across all *system* utterances

FIGURE 6.9: D-1 and D-2 distributions over all user and system utterances

		D-N	Spearman-ρ	Kendall-τ	D-N 2017
1-gram	user	0.73	0.0769	0.0568	0.73
	system	0.26	-0.0309	-0.0224	0.31
2-gram	user	0.82	-0.0404	-0.0299	0.80
	system	0.70	-0.0430	-0.0314	0.74

TABLE 6.5: Average Distinct-N on corpus level for both user and system. 2017 Distinct-N is also listed for comparison.



(A) (B)

FIGURE 6.10: Comparison between user and system lexical diversity based on 1- and 2-grams for both 2017 and 2018 systems.

	NEs per turn		NEs per session	
	User	**System**	**User**	**System**
mean	0.17	1.14	3.04	24.10
std	0.16	0.83	4.05	33.38
min	0.00	0.00	0.00	0.00
25%	0.05	0.53	1.00	4.00
50%	0.15	1.12	2.00	13.00
75%	0.25	1.63	4.00	31.00
max	5.00	24.25	104.00	769.00

TABLE 6.6: Named Entities per turn and per session mentioned by the user and by the system across all turns in the 2018 corpus.

6.5 Named Entities Analysis

Given the addition of the OntologyBot, ClarificationBot, and RedditBot to the ensemble, being bots that use and generate NEs, a further investigation is required to determine whether or how much the addition of extra bots under this (Information Retrieval) category influences the ratings, but also whether it instigates the user to use even more NEs, leading to more diverse conversation topics.

To that end, the same approach as described in Section 4.5 was followed to calculate the number of NEs mentioned by both the user and the system separately. From the findings presented in Table 6.6, it seems that actually in 2018 the users used on average 0.03 *fewer* entities per turn and 0.56 *fewer* per session. Likewise, the system in 2018 used 0.37 fewer Named Entities per turn on average and 8.82

fewer per session. However, the ratings in 2018 (as presented in Figure 6.11) follow the same linearity and direction as in 2017 with regards to the number of NEs mentioned.

Regarding the number of entities mentioned by the user, Figure 6.11 shows the same trend as in 2017, however (as more clearly shown in Figure 6.12), in 2018 there is a more steady and less variant increase of the trend of ratings with regards to the number of Named Entities mentioned. This however could be attributed to a number of factors, e.g. the quality of the NLG templates constructed in 2018 in these bots.

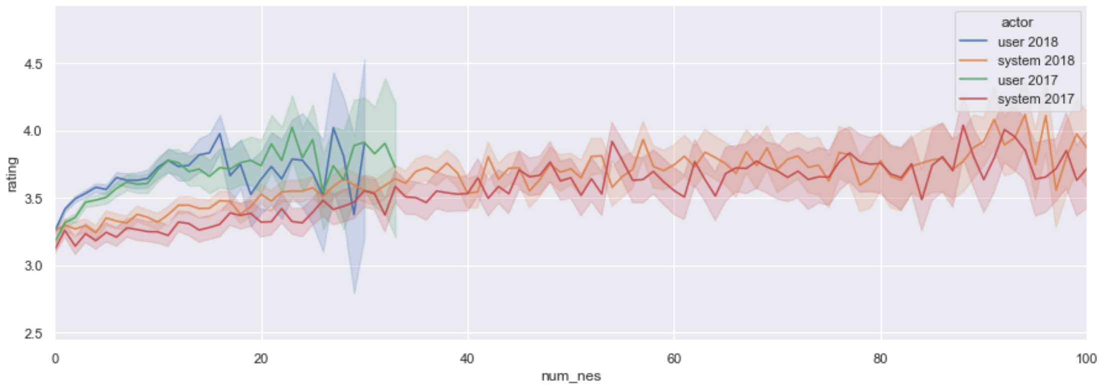


FIGURE 6.11: Relation between number of NEs and dialogue rating for both 2017 and 2018. Axis x shows the total number of NEs per session mentioned by the user and the system. The faded areas denote aggregation over multiple values on the y axis at each value of x and shows an estimate of the central tendency.

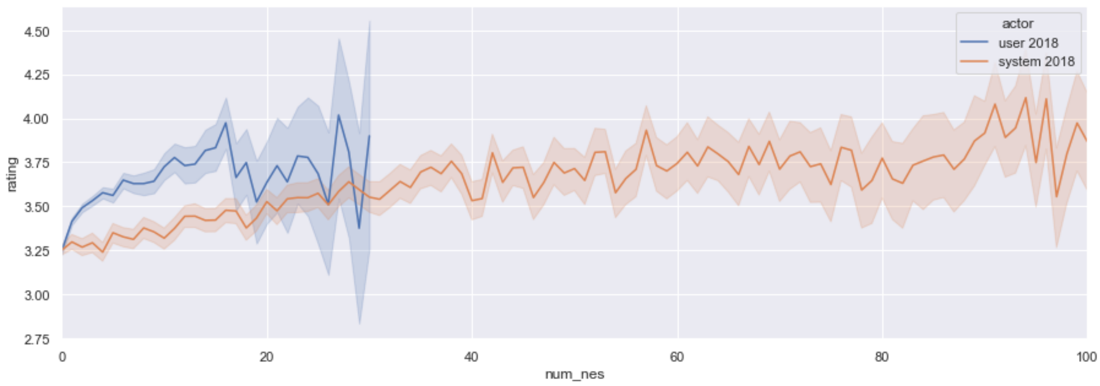


FIGURE 6.12: Representation of the relation between number of NEs and dialogue rating for 2018 only.

	Spearman-ρ	Kendall-τ
user	0.0882	0.0692
system	0.1073	0.0810

TABLE 6.7: Correlation between max number of Named Entities mentioned per session and dialogue ratings.

	sample average rating	**sample std**	**median**	**sample size**	**Z**	**p-value**
S_{wiki_1}	3.74	1.32	4	712	4.6601	**1.433e-06**
S_{wiki_0}	3.32	1.53	4	712		
S_{news_1}	3.69	1.33	4	712	4.0549	**2.8183e-05**
S_{news_0}	3.32	1.53	4	712		
S_{eliza_1}	3.50	1.40	4	575	-0.0902	0.9259
S_{eliza_0}	3.49	1.45	4	575		
$S_{persona_1}$	3.62	1.37	4	474	1.5814	0.1027
$S_{persona_0}$	3.45	1.46	4	474		
$S_{ontology_1}$	3.74	1.26	4	979	5.0580	**1.8277e-07**
$S_{ontology_0}$	3.36	1.49	4	979		
S_{reddit_1}	3.68	1.41	4	984	2.8805	**0.0028**
S_{reddit_0}	3.49	1.45	4	984		
$S_{coherence_1}$	3.52	1.44	4	1000	6.0961	**3.5238e-10**
$S_{coherence_0}$	3.07	1.60	3	1000		

TABLE 6.8: Two-tailed Mann-Whitney U test on the 2018 dataset examining the presence vs non-presence of the different bots. Due to the large sample sizes the significance Z is calculated using the formula 4.1. Bolded p-values denote significance.

6.6 Hypothesis testing of bot significance

Similarly to Section 4.6, for the 2018 dataset, the Mann-Whitney U was employed across the same (but now updated) bots used in the 2018 version of the system (WikiBot, NewsBot, ElizaBot, PersonaBot), alongside with tests of the newly developed bots introduced that year: OntologyBot, and RedditBot. Due to the changes in functionality after the consolidation of the 2017 versions of CoherenceBot, IntroBot, and RapportBot into the 2018 CoherenceBot, the impact of this bot is also tested here.

The added null hypotheses tested here are:

- $H0_{2a}$: **The existence of NewsBot has no effect on the dialogue ratings**
- $H0_{2b}$: **The existence of WikiBot has no effect on the dialogue ratings**
- $H0_{2c}$: **The existence of ElizaBot has no effect on the dialogue ratings**
- $H0_{2d}$: **The existence of PersonaBot has no effect on the dialogue ratings**
- $H0_{2e}$: **The existence of OntologyBot has no effect on the dialogue ratings**
- $H0_{2f}$: **The existence of RedditBot has no effect on the dialogue ratings**
- $H0_{2g}$: **The existence of CoherenceBot has no effect on the dialogue ratings**

The samples where taken at random in the same manner as previously discussed as $N_{b0} = N_{b1}$, with $b \in [wiki, news, eliza, persona, ontology, reddit, coherence]$, being the sample of each of the tested bots. A positive (bot responded *at least in 20%* of the total system responses in that dialogue) and a negative (bot wasn't present in the dialogue) sample was collected for each of these bots from the AL2018 dataset. The Mann-Whitney U results are summarised in Table 6.8.

Out of the 7 bots that were tested in 2018, the presence of WikiBot, NewsBot, RedditBot, OntologyBot, and CoherenceBot had a significant ($p \leq 0.05$) impact on the dialogue ratings. Out of these five, CoherenceBot had the biggest impact (almost 0.5 difference on the sample's average on the 5-point Likert scale). Thus, hypotheses $H0_{2a,2b,2e,2f,2g}$ can be rejected.

This shows the significance of the features this bot encompasses (i.e. user profiling, topic shifting, conversational drivers, rapport building, etc - as described

	Spearman-ρ	**Kendall-τ**
WikiBot	0.1412	0.1096
NewsBot	0.0875	0.0690
ElizaBot	0.0886	0.0722
PersonaBot	0.1024	0.0844
OntologyBot	0.0467	0.0362
RedditBot	0.1712	0.1336
CoherenceBot	0.0682	0.0522

Table 6.9: Correlation between the number of responses per session of each bot and the dialogue's rating

in Section 5.2.3.8). However, it also emphasises a major design flaw in 2018's architecture, which was to combine all these features into a single bot, creating a single-point of failure. This is something that should be addressed in future work. It is also worth noticing that in 2018 ElizaBot's and PersonaBot's presence in the conversation was less significant than in 2017. An additional observation would be the good performance of the OntologyBot, which proves the fact that Entity Linking has a strong impact on raising the user score (3.36 to 3.74).

Following that, the correlation between the number of turns each of these bots handled in a dialogue with that dialogue's rating is calculated. The results are shown in Table 6.9. All bots show some weak positive correlation, although both WikiBot and RedditBot stand out with almost double the correlation compared to the rest. In the case of RedditBot, this can be attributed to the fact that the responses it generates are mainly of the humorous and entertaining kind, which were inherently designed to improve user engagement. WikiBot's correlation on the other hand, shows that creating an "all-knowing" agent (given the fact that this bot had the entire Wikipedia available as a knowledge source) can increase user ratings in an open-domain environment.

6.7 Hypothesis testing of bot version significance

Following the completion of the 2018 competition, further investigation was required to reflect upon the effect of the changes and optimisations of the most altered bots in the ensemble between the two competitions/versions, i.e. *NewsBot, WikiBot, PersonaBot,* and *CoherenceBot*. To that end, samples were collected between the AL2107 and AL2018 datasets of dialogues where each of the tested bots was present on at least 20% of the system responses in the dialogue. The samples had the same size across each bot pair as shown in the result Table 6.10. For the 2017 CoherenceBot sample selection specifically, conversations where any of the bots in CoherenceBot, IntroBot, or RapportBot had responded (more than 20% in the dialogue) were considered to be on par with the consolidated functionality of the 2018 version of CoherenceBot. Then the following null hypotheses were tested using the Mann-Whitney U test:

- $H0_{7a}$: **NewsBot in 2017 and NewsBot in 2018 had the same effect on the dialogue ratings**
- $H0_{7b}$: **WikiBot in 2017 and WikiBot in 2018 had the same effect on the dialogue ratings**
- $H0_{7c}$: **PersonaBot in 2017 and PersonaBot in 2018 had the same effect on the dialogue ratings**
- $H0_{7d}$: **CoherenceBot in 2017 and CoherenceBot in 2018 had the same effect on the dialogue ratings**

The findings in Table 6.10 show that no significance exists between the versions of each bot in terms of ratings. It is observed however that in most cases (excluding PersonaBot) the changes made to the bots in 2018 had a positive impact on the dialogue ratings.

	sample average rating	sample std	median	sample size	Z	p-value
WikiBot17	3.62	1.46	4	247	1.3473	0.1640
WikiBot18	3.75	1.35	4	247		
NewsBot17	3.61	1.32	4	712	-14.9514	0.3752
NewsBot18	3.69	1.33	4	712		
PersonaBot17	3.69	1.35	4	474	-0.7215	0.4546
PersonaBot18	3.62	1.37	4	474		
CoherenceBot17	3.42	1.46	4	1000	1.6957	0.0877
CoherenceBot18	3.52	1.44	4	1000		

TABLE 6.10: Mann-Whitney U test between the 2017 and 2018 version of the most changed bots, two-tailed. Due to the large sample sizes the significance Z is calculated using the formula 4.1. Bolded p-values denote significance.

6.8 Correlation between ASR and dialogue rating

Finally, the 18,765 ASR annotated dialogues were examined for correlation between the confidence score and the dialogue's rating. The calculated Spearman-ρ was found at 0.0218 while Kendall-τ was found at 0.0160, leading to no correlation between the tested variables, as further portrayed in Figure 6.13.

It is worth noticing though that the confidence scores received from the ASR did not carry enough information to be useful. Open-domain conversation would be better optimised with an ASR where the confidence score is more informative (e.g. hesitations and disfluencies annotation, as further discussed in Chapter 8).

6.9 Observations and Discussion

The severe system overhaul in 2018 resulted in a more steady and reliable system that was able to scale better with the number of calls the system was receiving each day. However, the system struggled momentarily from a sudden increase of calls (from ~ 500 to ~ 2000 – due to Amazon developers being invited again to interact with the socialbots to test their performance and capabilities), although

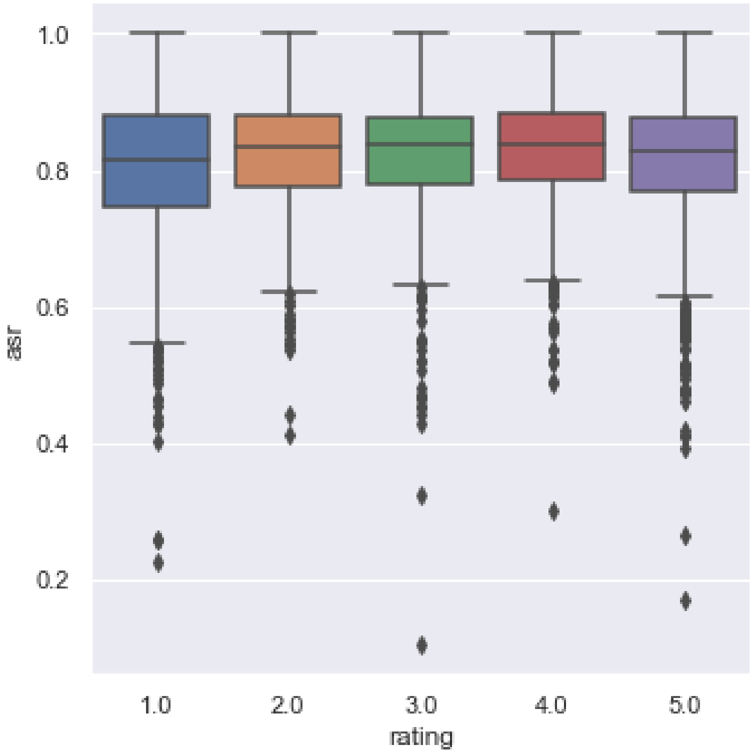


FIGURE 6.13: Boxplot between average ASR confidence score per session and rating. A sample 20% of the annotated dataset is visualised for better readability.

this issue can be avoided with proper DevOps background knowledge that the team lacked at the time.

The ratings in 2018, apart from the aforementioned dip in October, varied less than in 2017 and had a steady mean of around 3.6 in the 5-point Likert scale. Additionally, due to the nature of the competition, the uncontrolled group of users, and the extremely sparse signal (single rating at the end of the conversation) resulted in this plateau effect making it almost impossible to increase the average score over 3.8-4.0. This is also supported by the average scores of the winning teams of the Alexa Prize Challenge in 2019 (3.81) and 2020 (3.28) received.

Regarding the correlation results, they remained mostly unchanged from those of the previous year. However, a lot of the changes made to the 2018 system were in the back-end of the system (e.g. increasing modularity, scalability, etc) so their contribution is not visible to the end-user. Additionally, although the correlations were found to be weak, the changes made to each individual bot increased their significance to the overall user rating, as summarised in Table 6.11.

	p-value (2017)	**p-value (2018)**
S_{wiki}	0.0016	1.433e-06
S_{news}	1.0753e-05	2.8183e-05
S_{eliza}	0.3847	0.9259
$S_{persona}$	0.0018	0.1027
$S_{ontology}$	N/A	1.8277e-07
S_{reddit}	N/A	0.0028
$S_{coherence}$	N/A	3.5238e-10

Table 6.11: Summarised comparison of the significance of bots in 2017 and 2018. Data aggregated from Tables 4.8 and 6.8.

Chapter 7

Combining task-based and social conversation in an embodied conversational robot

Robots have been used for a number of years for various different tasks (from industrial applications to social companions) and with the advent of social robotics in the last several years there is an increasingly larger need to make Human-Robot Interaction (HRI) feel as natural as possible as people get used to existing with them in their daily lives. Additionally, as argued in Papaioannou and Lemon (2017), Papaioannou et al. (2017b), an assisting agent in the shopping mall domain should also be entertaining. To that end, we wanted to add to Alana extra modalities that transcends that of a voice-only social interactive agent.

This chapter focuses on addressing two main issues: (a) how can a combination of task-based and 'social' open-domain dialogue system be designed? (b) How does adding an embodiment to a spoken dialogue system affect the user's perception and overall rating towards it?

An integrated system using Alana and an embodied robotic platform is presented, as well as the design decision motivations and challenges this presents, addressing

Research Question 5 (see Section 1.2). Finally an evaluation of the integrated system during a long-term-deployment in a shopping mall in Finland is discussed.

7.1 Uni-modal vs Embodied Multi-modal interaction

The work presented here so far focuses on uni-modal dialogue systems alone (text or voice input). However, humans make extensive use of non-verbal cues where many of them are subtle and even involuntary such as stance, gaze, and gesture (Cassell and Vilhjálmsson, 1999). Consequently, many scientists are working on developing natural and appropriate nonverbal signals for Embodied Conversational Agents (ECAs), as well as recognising and processing signals produced by users (Weiss et al., 2015).

An ECA is an agent that is capable of interacting with users using not only Natural Language, but also human-like features (e.g. gestures, facial expressions) and/or body parts. These ECAs can be grouped into two main categories:

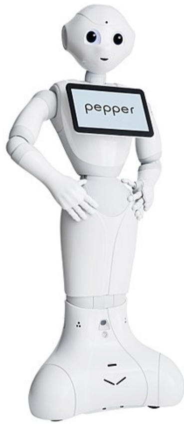
- **Virtual embodied CAs**, where the CA is a graphical model of a humanoid.
- **Humanoid Robotic Agent**, where the CA is running on a human-like robot.

The virtual embodied Conversational Agent (CA) may consist of only limited parts of the human body (e.g. torso only, or face only), or it may use an entire human-like body to extend its interaction, as shown in Figure 7.1a. These types of CAs although easier to create most of the time, suffer from the *Mona Lisa effect* (Sato and Hosokawa, 2012), making it hard for the user to understand what the CA is actually gazing at. This gives a less natural feeling during HRI.

A humanoid robotic CA on the other hand, might not suffer this effect. However, due to the fact that they are a lot more complex and difficult in production, they

(A) Various virtual agents for sign language[a]

[a]Image retrieved from https://achrafothman.net/site/virtual-conversation-agent-avatar-for-sign-language/



(B) Softbank's Pepper robot

FIGURE 7.1: Different types of ECAs

don't always succeed in imitating human gestures, gaze, and movements as fluently as their counterparts.

The goals and challenges addressed here were tackled through the Multi-Modal Mall Entertainment Robot (MuMMER) project. This project aimed to produce a robot that was both useful and entertaining. Therefore, it had to support tasks as well as social conversation capabilities. The aims of this reseach were to investigate:

- how to combine task-based dialogue with social dialogue
- how embodiment affects the user ratings compared to a voice-only modality
- developing approaches to the implementation of a scalable and robust multi-modal social system
- exploring the porting of Alana's language (into Finnish)

7.2 The MuMMER Project

MuMMER was an EU-funded 4 year joint project between the University of Glasgow (GLA), Heriot-Watt University (HWU), Idiap Research Institute (IDIAP), LAAS-CNRS (LAAS), VTT Technical Research Centre of Finland (VTT), Softbank Robotics Europe, and IdeaPark (IDP)[1]. During this project, a social robot was designed to interact with users in a shopping mall (Ideapark, Finland) in a natural and flexible way and in the users' native language - *Finnish*. SoftBank Robotics' Pepper humanoid robot was employed as the deployment platform for this project (Foster et al., 2016). The developed system includes conversational interaction, geometric reasoning, audio-visual sensing, perspective taking, social signal processing, and motion planning (Foster et al., 2019). The different components were combined into a framework using the Robot Operating System (ROS)[2].

7.2.1 Project Description and Architecture

The main focus of the MuMMER system was to combine a task-based dialogue system with open-domain social interaction, in order to fulfil user tasks, such as finding a particular shop or product, while engaging in natural, entertaining, and engaging conversation with them.

In the MuMMER project, Alana is the core module for any dialogue interaction with the user from every other module in the project. This means that Alana can handle any task in which a module needs to either verbally alert the user or obtain input from the user. Such tasks can include asking directions to a particular shop, where the nearest accessible toilets are, which shop sells a particular item, etc. As a result, the dialogue should be contextually meaningful and easy to sustain during the interaction. A challenge addressed here was the fact that most tasks required multiple turns to complete and have a certain duration. This leads to increased probability of the user interrupting an ongoing task, changing or cancelling the goal, or even requesting additional tasks from the system. Alana was therefore augmented with so-called *task bots* in its ensemble of bots (as previously shown in Section 5.1) to conversationally conduct and track behaviours on a physical

[1] http://mummer-project.eu/partners/
[2] https://www.ros.org/

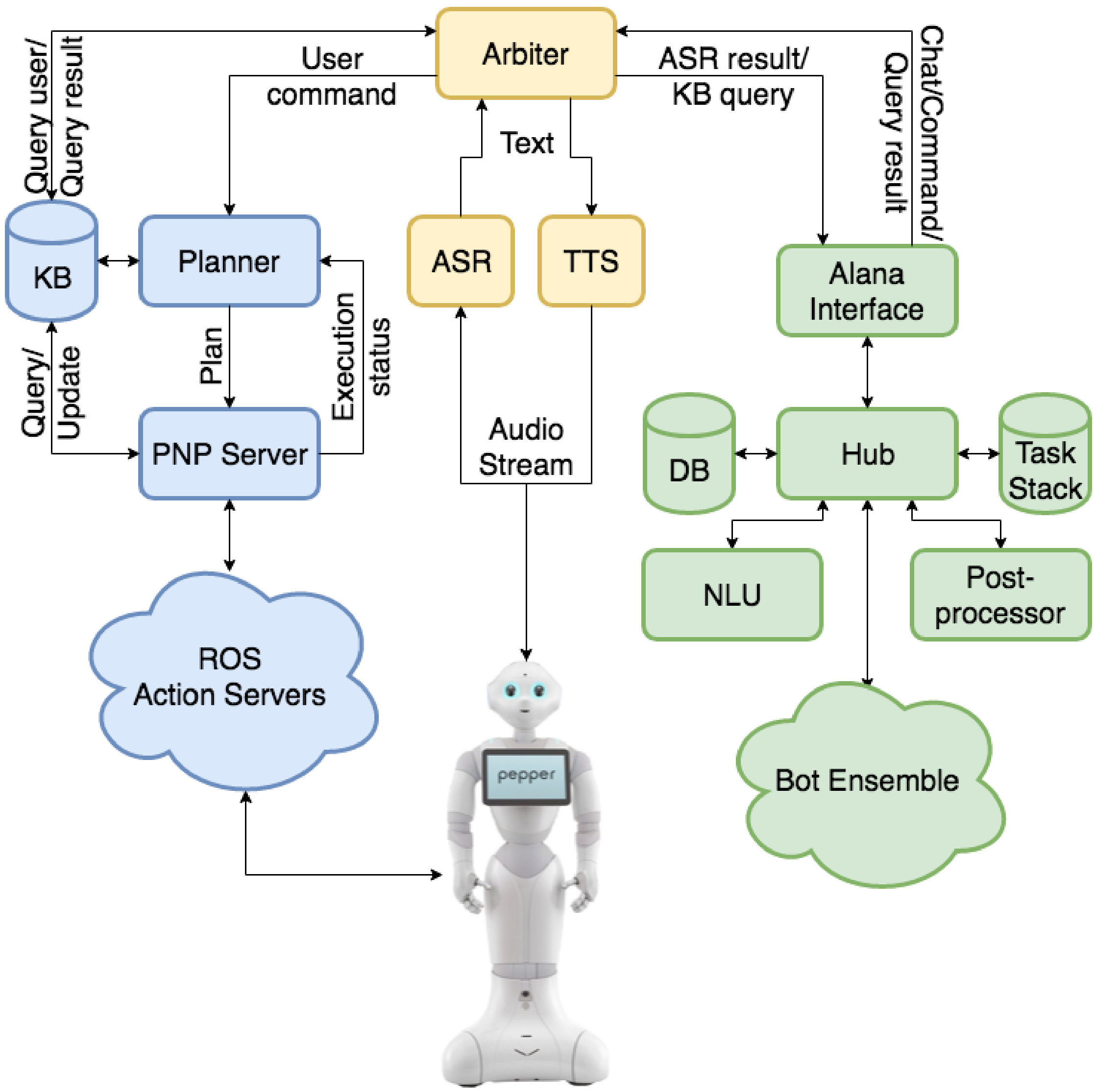


FIGURE 7.2: Architecture of the MuMMER system. The blue part denotes the task management and execution system, whereas the green part denotes Alana as the dialogue system. Alana's Bot Ensemble contains social chat bots and a *task bot* that is able to perform tasks while handling the communication between a task and the user. The yellow part in the middle includes the TTS and ASR, and an integration component (Arbiter).

agent, as the robot needs to participate in social dialogue as well as complete tasks (Figure 7.2) (Papaioannou et al., 2018).

A new NLU module, HERMIT NLU, trained on the ROMULUS corpus (Vanzo et al., 2019), was introduced and incorporated into the Alana framework to allow the functionality mentioned above. It is capable of dealing with social chit-chat as well as extracting the required information from commands to execute tasks.

user: Where can I find a coffee shop?
system: There are two close by. Do you prefer Shop_1 or Shop_2?
user: Oh, I never asked for your name! What are you called?
system: My name is Alana! Nice to meet you. So, about the coffee shop. Which one would you prefer?
user: Cool! Anyway I prefer Shop_2.
system: Alright! Directions to Shop_2. Do you see the red sign on the left? ...

FIGURE 7.3: Example multi-threaded dialogue (Papaioannou et al., 2018)

HERMIT NLU is thus used to determine if a task-related utterance (e.g. *"Where is Minimani*[3] *?"*) has been produced so the task bot can be triggered, and to collect task parameters (e.g. the name of the shop someone is looking for).

7.2.2 TaskBot: Task-handling in MuMMER and multi-threaded dialogue

Following the identification of a task, it is normally carried out using physical actions on the robot that take a finite period of time to complete, as opposed to dialogue actions which are usually rapidly completed by TTS. One of the challenges here is that the user may want to continue a non-task related conversation or send new instructions while the robot is performing a certain task, for example moving to pointing to a specific location. An initial approach to combine task-oriented and social dialogue was to employ RL to train a policy for the Alana's DM to be able to take actions $a_t \in A$ where A = [*PerformTask, Greet, Goodbye, Chat, GiveDirections, Wait, RequestTask, RequestShop*]. This initial work is presented in Papaioannou et al. (2017b). It was soon decided, however, that this approach did not cater for multiple or concurrent tasks present in a single dialogue. Thus, the notion of *recipes* presented in Lemon et al. (2002) is heavily employed in order to implement a multi-threaded conversation management pipeline, interleaving multiple tasks with general chitchat. The functionality of these recipes is explained in more detail later on in Section 7.2.3.

[3]Minimani is a Finnish shop present in the Ideapark shopping mall.

In order to achieve this objective, the execution system introduced in Dondrup et al. (2017) was extended to use the above mentioned recipes which define the dialogue and how to execute physical actions (Papaioannou et al., 2018). The framework described in Dondrup et al. (2017) was redesigned to support multi-threaded execution and a process of arbitration was also established to administer the current tasks via the Alana dialogue system. To that end, a new type of sub-bot was added to the ensemble - *TaskBot*, which is able to interpret the recipes for different tasks, as described below. This enables tasks to be started, halted, or stopped, while providing feedback to the user at appropriate times (Foster et al., 2019). While the newly developed TaskBot keeps track of the history and the current state of the running tasks, i.e. the last question raised by the system, the Arbiter (see Section 7.2.3) keeps track of all ongoing tasks and their relationships to potential responses. This allows the system to pause and resume any task so that the user can interrupt, for example, a guidance task by asking the user about their day, and then resume the guidance task. This functionality is supported by Alana re-raising previously unanswered system questions (Figure 7.3).

These new MuMMER-specific sub-bots within the ensemble which provide task-related dialogue responses exchange information with the planning and execution framework using the Arbiter through the incorporation of specifically formatted structure (*recipes*) in the reply for task-related conversation turns. These recipes contain all the information required for the next dialogue task, in addition to text that will be synthesised through TTS (as opposed to the social dialogue sub-bots that require and respond using text messages only). Using these recipes, the conversational system is agnostic to the fact that the system's response includes planner commands. This results in the retention of all task and non-task related information as a shared dialogue context by treating planner commands and responses equally, free to be used by any other bot in the ensemble. Additionally, the design of the TaskBot and the recipes' structure was done in such a way to accommodate the execution of different tasks by simply writing a new recipe describing the task. This allows the easy extension of the systems when additional tasks were included (e.g. adding simple quiz-games or invoking the robot's build-in dancing and other skills). Furthermore, these recipes and their consumption by the TaskBot allowed the integration of the different task-related services and components in the MuMMER architecture developed by the different partners. For example, the *area Y* mentioned in Fig. 7.4 or the fact that the route includes *stairs* are retrieved from an ontology built by LAAS for the purposes of this project. This is actualised by the TaskBot initially informing the Arbiter about the information gathered from the user's utterance (using the recipe code `task_route_descr.code` as shown in the recipe example below).

Any task that a user instructs the robot to carry out results in the creation of a plan that may involve multiple actions and nested sub-actions to be carried out in order to achieve the desired result. These behaviours can also necessitate intermediate user feedback to explain or disambiguate such requests, all while allowing the user to delay or cancel the whole plan at any time. As a result, more than one dialogue turn may be needed to successfully complete a task, raising the question of how to create a system that is scalable and enables the user to not only interleave multiple tasks as in Lemon et al. (2002), but also keep the user engaged while executing the planned task (Fig. 7.3).

In the example in Figure 7.3, while the user starts the conversation with a clear intention of finding a coffee shop, they quickly get side-tracked from that goal into a more social style of interaction. At that point the implemented system not only coherently responded to the sudden change of the task-irrelevant turn, but also managed to re-engage the user into the task by re-raising the last system's request, allowing the active task to continue. To that end, the system needs to be able to effectively handle conversations in a multi-threaded way (fig. 7.4), keeping track of active tasks, re-engaging with the user on a particular task, and being able to pause and resume tasks accordingly.

Although there is promising work in this field (Heeman et al., 2005, Kun et al., 2010, Lemon and Gruenstein, 2004), the main goal here is to build a complete and scalable framework that facilitates the swapping between multiple task "threads" and non-task dialogue in a more natural and human-like manner. According to

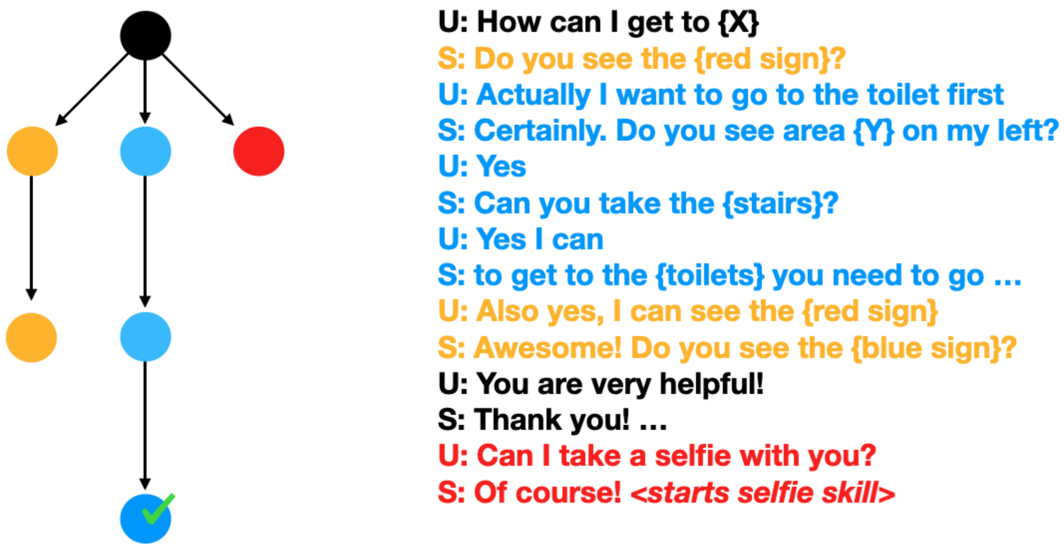


FIGURE 7.4: Example of a multi-threaded dialogue. The final ticked step denotes the plan completion and closing of an active thread.

Lemon and Gruenstein (2004), on each turn, when a task-related utterance is identified, the system first checks to see whether it resolves an active and incomplete task thread (e.g. the user gives an answer to a system's query needed for the completion of a task - e.g. *S: "Can you take the stairs?" U: "Yes"*) in which case it can be added to an existing task thread, or whether it must initiate a new one. All of the tasks are stored in order of recency in a *Task Stack*, and if a task is finished, it is removed from the stack. The *Task Stack* keeps comprehensive status details for each task, such as unmet preconditions, unanswered questions to the user, and so on. Some tasks in the stack are also prioritised, allowing them to circumvent the recency order in the sense that they should be deemed more urgent and completed first (e.g. the robot is running out of battery, robot has tipped over and requires human help, etc.).

This method also allows the user to interrupt a specific task while the robot is performing it (e.g., "stop it"). In this case, Alana's co-reference resolution functionality will be able to convert it into a meaningful task identifier and send an interrupt command to the Arbiter node. More specific statuses can be given in the same way (e.g. "What are you doing now?") as in Lemon et al. (2002), where the system can respond to the user in a natural and coherent way, ideally "echoing" the user's language (e.g., "I am guiding you to Shop 2," assuming an interaction similar to that shown in Fig. 7.3).

7.2.3 The Arbiter

As mentioned earlier, Alana is the single module in the MuMMER architecture that handles all dialogue specific requests from any sub-module in the pipeline when verbal interaction with the user is required. To that end, an 'Arbiter' has been introduced as a conduit and translator between the different modules and the Alana framework as shown in Figure 7.2. The Arbiter decides whether Alana's output is to be sent to the Planner or TTS module. For example, the user might ask *"Where can I find Costa?"* which will trigger a task-specific intent in the NLU which will be picked-up by the task-bot to resolve the task by consulting the recipe (see below in this section for an example). The task-bot would then generate the following response, which will be identified by the Arbiter as an output to be sent to the Planner.

```
{
 'action': 'task_route_descr',
 'params': {
    'place_frame': 'Costa', 'person_frame':
   'human-03fe897a-2145-41ef-ab81-6dc22f4098f1'
 },
 'confirmation': 'OK',
 'command': 'execute',
 'task_id': '74d6af8f-4fb7-4653-aaac-b111b4c8e76b'
}
```

As seen in the example below, the tasks can be represented using YAML markdown. These may vary from simple verbalisations related to tasks (for example, lines 15-19) to more complex, multi-turn mini-conversations like clarifications (lines 5-14). In the *route_repeat* example below, the system asks the user *"Should I show you the direction again?"* and expects the user to confirm or deny that prompt in order to consider the overall task completed. The example recipe below is for a route description task that is instructing the system to ask the user whether they wish to be reminded of the route steps at the end (`task_route_descr.status.route_repeat.return_tts.text`). The `verbalisation` part is requested by the Planner if the user remains silent after the system asked *"Should I show you the direction again?"*.

```
task_route_descr:
  code: '{{"action": "{intent}", "params": {{"place_frame":
   ↪  "{param}", "person_frame": "{user_id}"}},
   ↪  "confirmation":"{confirmation}", "command":"execute",
   ↪  "task_id":"{task_id}"}}'
  confirmation: 'OK'
  status:
    route_repeat:
      resolve:
        YES: "^(?:(fine|yes|yeah|yea|yep|aye|okay|ok|sure)(
         ↪  (fine|yes|yeah|yea|yep|aye|okay|ok|sure))*( please)?|
         ↪  ((yes|yeah|yea|yep|aye|ok|okay) )*(sure thing|i guess(
         ↪  so)?|go ahead|i would like
         ↪  that)|((yes|yeah|yea|yep|aye|ok|okay)
         ↪  )*(i|you|he|she|it|we|they)
         ↪  (do|does|am|is|are|have|has))"
        NO: "^(?:(i do not|no|nah|nope)(
         ↪  (no|nah|nope))*|((no|nah|nope)
         ↪  )*(?:absolutely|certainly|of course|probably|i guess)
         ↪  not|no way|((no|nah|nope) )*(i|we) do not want
         ↪  to((no|nah|nope) )*(i|you|he|she|it|we|they)
         ↪  ((do|does|am|is|are|have|has)(n't| not)|ain't))"
        return_tts:
          text:
            - "Should I show you the direction again?"
```

```
        return_cmd: '{{"action":"{intent}",
          ↪  "params":{{"place_frame":"{param}"}},
          ↪  "return_value":{result},
          ↪  "confirmation":"{confirmation}", "command":"execute",
          ↪  "task_id":"{task_id}"}}'
        confirmation:
          - "okay"
        verbalisation:
            get_attention:
                return_tts:
                    text:
                        - "Hey, are you listening ?"
```

7.2.4 Porting to a different language

For the purposes of this project, the ROMULUS corpus (Vanzo et al., 2019) was created which provided the training examples for the HERMIT NLU and was annotated with DAs (Stolcke et al., 2000) and Frame Semantics (Fillmore, 1985). ROMULUS is composed of 1,431 English sentences (Table 7.1), for each of which dialogue acts, semantic frames, and corresponding frame elements are provided. For example, for the sentence *"Where can I find Minimani?"* expresses a LOCATING frame, with the verb *find* playing the role of LEXICAL_UNIT. In this example the frame elements are *I* (PERCEIVER) and *Minimani* (SOUGHT_ENTITY) as explained in Vanzo et al. (2019). This dataset was created to model user utterances to open-domain conversational systems for robotic systems that will be used in a variety of situations, for example in chit-chat or command interpretation.

As mentioned earlier, one of the requirements of the MuMMER project was for the robot to be able to interact in Finnish. Two approaches were considered to develop this functionality. The first solution was to re-develop the whole Alana conversational system in Finnish. This solution requires expertise in the Finnish language (which we lacked within HWU - the consortium partner responsible for the conversational capabilities of the system), as well as huge refactoring of all datasets, knowledge bases, and re-training/refactoring the NLU models to that language since Alana's NLU trained modules, rules, and patterns are implemented for the English language. The solution we explored instead was to employ Machine Translation (MT) to obtain the English sentence from the Finnish one and to translate all of Alana's English NLG outputs to Finnish using MT. This allowed us to focus on the English system, while retaining the ability to "understand" Finnish. To that end, in the Finnish version of the system, the user's utterance was automatically translated into Finnish using Google's Cloud Translation API[4], then the (now in English) utterance was allowed to pass through Alana's NLU pipeline as it normally would. Of course, MT is not able to provide gold translations and the error of the translation is propagated throughout the processing.

To better understand this, consider the user's sentence "*Kummassa päädyssä on Minimani?*" which means *"Where is Minimani?"*. However, Google's MT translates this sentence into "*Which end is Minimani?*", where semantics are hardly parsable by any grammar-based semantic parser. Another example is when the users wanted to take selfies with the robot (one of the - very popular - functionalities of the system). However, in this case, the request to take a selfie together "*Voiks sä tulla mun kanssa selfieen?*" would be translated to "*Can you come with me selfie?*", which provides completely erroneous user intent. To address this issue, apart from (a) relying on the statistical nature of the NLU models that allowed us to generalise on unseen data, we (b) extended the ROMULUS corpus by including 167 automatically translated sentences (Finnish to English), to model the consistent errors made by MT. HERMIT NLU was then retrained on the updated ROMULUS corpus. These two solutions allowed us to improve the performance of the NLU system when deployed in the real shopping-mall scenario.

Although the user's utterance was automatically translated, some of the system's responses were manually translated to Finnish by native speakers. We experimented with automatic translation on the responses as well, however Google's

[4]`https://cloud.google.com/translate/docs/reference/rest`

	ROMULUS dataset
Sentences	1431
Sentences length	7.24
Dialogue act labels set	11
Frame labels set	58
Frame element labels set	84
Number of dialogue acts	1906
Number of frames	2013
Number of frame elements	5059
Dialogue act/sentence	1.33
Frames/sentence	1.41
Frame elements/sentence	3.54

Table 7.1: Statistics of the ROMULUS dataset.

API translation was deemed too unreliable and found lacking naturalness for native Finnish users. It's worth noting here that only the system's template-based responses were manually translated that way, and not the automatically generated or information-retrieved responses (such us responses generated by the *WikiBot* or *NewsBot*), since these responses would have to be automatically translated, leading to the aforementioned problems. Furthermore, at the time, finding and indexing Finnish knowledge sources was not a viable option due to time and resources constraints. Consequently, the information retrieval bots were disabled in the Finnish version of the system, leaving only the rule and template based ones active, providing full control over the Finnish responses. The final ensemble of bots deployed in the Finnish version were *PersonaBot* (to provide rule-based social dialogue), *CoherenceBot* (for user modelling, topic shifting, etc as presented in Section 5.2.3.8), and *TaskBot* (for performing tasks in the shopping mall domain).

In the example below, in the equivalent Finnish version only the *return_tts.text* and *confirmation* parts of the recipe are translated. We also found that although some responses remains the same in both languages (e.g. *"OK"*), using the Finnish spelling sounded more natural using the Finnish Google's TTS api.

```
        NO: "^(?:(i do not|no|nah|nope)(
         ↪  (no|nah|nope))*|((no|nah|nope)
         ↪  )*(?:absolutely|certainly|of course|probably|i guess)
```

```
      not|no way|((no|nah|nope) )*(i|we) do not want to((no|nah|nope) )*(i|you|he|she|it|we|they) (
      (do|does|am|is|are|have|has)(n't| not)|ain't))"
    return_tts:
      text:
        - "Näyttäisinkö suunnan uudelleen?"
    return_cmd: '{{"action":"{intent}", "params":{{"place_frame":"{param}"}}, "return_value":{result}, "confirmation":"{confirmation}", "command":"execute", "task_id":"{task_id}"}}'
    confirmation:
      - "okei"
```

7.3 Experiment: MuMMER Long Term Deployment and Findings

The final evaluation of this four year project was performed through a long-term deployment[5]. The experiments had a variety of goals. One of the most important aspects to investigate was the users' perception of the robot's usefulness, engagement, and entertainment value. This is a necessary precondition because the robot was intended to be a service provider. Another critical aspect is the effectiveness of the proposed techniques in how to deal with various problems. Such analysis could provide additional support to the project's research outputs. Finally, the long-term deployment allowed us to evaluate the system's robustness when deployed in the field in a real life scenario. This provides an assessment and great insight of the engineering solutions that enabled the overall system's design and development.

[5]The deliverables of this project are described in more detailed in http://mummer-project.eu/outputs/deliverables/

FIGURE 7.5: MuMMER robot deployment in the mall

This 14-week long-term deployment took place at the Ideapark shopping mall in Tampere, Finland (from 9th September to 13th December 2019). The robot was housed in a custom-built booth designed to isolate the platform from background noise and bright light. VTT and IDP operators were physically present to protect the robot from malintent while it was free to interact with visitors with the rest of the partners providing support remotely by fixing issues and restarting components as needed.

In the first weeks, all modules were mostly integrated, and major problems preventing the stability of the robot behaviour were resolved. Under the supervision of VTT and IDP, we then focused on tuning the Finnish system by identifying edge cases and fixing translations to increase the naturalness and appropriateness of responses when interacting with real users. The user studies and system evaluation began in week 9 when the system was reasonably stable for use in the field and ended in week 14. To that end, all of the statistical analysis that follows refers to weeks 9 through 14.

7.3.1 Empirical Evaluation

The analysis of data collected during long-term use is described in this section. During this period, all user interactions were automatically logged and users were asked to complete an optional questionnaire at the end of the interaction. The latter aimed at a more subjective assessment of the system and statistical analysis of qualitative data by collecting **56 questionnaires** in total during that time. Due to the various tests performed by the on-site researching team during the long-term-deployment it was impossible at the time to differentiate which logged dialogues took place with the researchers and which with actual participants. It is also worth noticing that the overall user rating question as part of the survey was the same as the one used during the Amazon Alexa Prize (*"Would you want to talk to this social bot again in the future?"*) to remove the question bias and be able to compare the results with those versions of the system.

7.3.1.1 User ratings and embodiment effect

This empirical study uses the Alexa Prize results of Alana as a baseline. One of the aims of the optional questionnaire given to the interacting users was to get feedback on the interaction experience with the robot. These scores range in a 5-point Likert scale, from 1 (being the lowest) to 5 (being the highest) . This corresponds to the user ratings gathered during the Alexa Prize, which were also on a scale of 1 to 5.

Figure 7.6 shows the average user rating of the MuMMER system compared to that of the Alexa Prize Alana 2017 (Papaioannou et al., 2017a) and 2018 (Curry et al., 2018) systems. In terms of user rating, the graph shows an upward trend. The increase between the Alana 2017 and 2018 systems is mostly due to all of the improvements made to enhance the conversational agent's efficiency and quality of interaction (as previously described in Chapter 5). However, it is interesting to analyse the MuMMER dialogue agent's continued progress, especially given the fact that several Alana bots have been disabled. The dialogue agent, therefore,

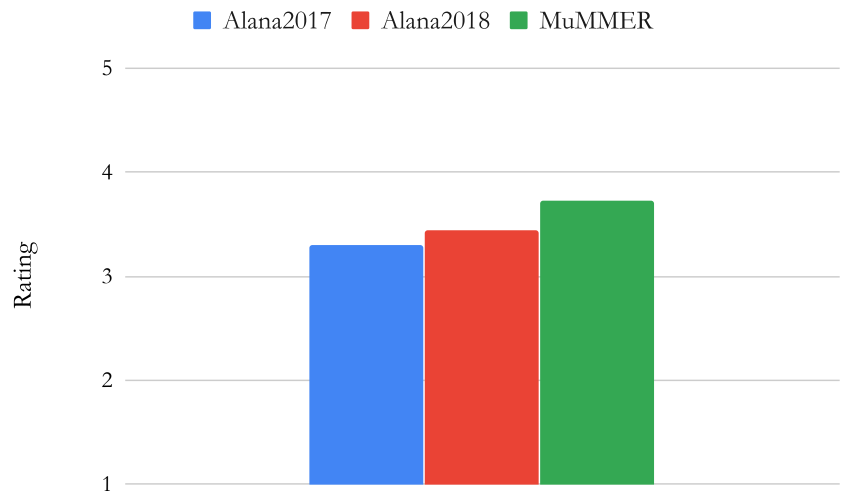


Figure 7.6: Average user ratings comparison.

does not cover all of the topical variations of the English version. We assessed that a) the embodiment of the physical robot and b) the MuMMER system's task usefulness can play a role in the increase of rating. Indeed, as illustrated in the literature (Ventre-Dominey et al., 2019), the incorporation of a robot into an interactive system can greatly increase its acceptability and likeability. Furthermore, the MuMMER system's ability to provide relevant directions and assist users with navigation and realistic search tasks likely leads users to score it higher than a just entertaining and social system like Alana.

It is worth noticing though that the user sample size between the Amazon Alexa evaluation and the MuMMER evaluation vastly differ (thousands of users in the former and only 56 in the latter). The two systems were also different in several other respects apart from the added embodiment in the MuMMER version, such as language (English versus Finnish) as well as the bots present in the ensemble of each version (all bots stated in Section 5.2.3.1 for 2018 and 3.3.3 for 2017 versus only PersonaBot, CoherenceBot, and TaskBot for MuMMER). This means that direct comparison cannot be made between the systems.

7.4 Conclusion

This chapter discussed the effect of embodiment and integration of task-based with social dialogue for a conversational agent (like Alana). The empirical study conducted during the MuMMER project showed that the combination of social conversation, task-performing capabilities, and an embodied agent was effective and useful for visitors in the shopping mall.

The main contribution of this research was the integration of task-based dialogue, robot planning, and social dialogue through the usage of recipes. The main challenges of multiple task handling and task concurrency in the same interaction were approached using the notion of *recipes* and *multi-threading*, allowing the user to pause, resume, restart, and cancel any current tasks as well as starting new tasks in parallel with any previous ones (as shown in Figure 7.4).

During the long-term-deployment evaluation of the integrated system, it was found that the system scored higher on the user ratings compared to the Amazon Alexa Prize 2017 and 2018, however the systems are not directly comparable due to a number of different confounding factors (such as different user sample size, bot ensemble differences, etc).

During this project, my main contribution was primarily focused in the design and implementation of the TaskBot, extending of Alana's capabilities to also perform tasks initially by employing a trained policy using RL. Following that, I used a different approach to tackle multiple and concurrent tasks in a multi-turn way with the use of recipes. Additionally, with the help of VTT, all NLG outputs were translated to Finnish.

Chapter 8

Conclusion and Future Work

8.1 Discussion

In this thesis the challenges and current methods of designing, building, and evaluating open-domain conversational AI systems were discussed. Different architectures for building conversational agents were described (modular, rule-based, neural, end-to-end, and ensemble) as well as the advantages and disadvantages each of these present in various conversational environments (task-based, open-domain). It was discussed that although individual challenges have been addressed and good progress has been made in recent years by the research community, there is not enough progress on system architecture and evaluation that is able to tackle all the challenges of open-domain conversation at the same time.

In Chapters 3 and 5, an approach addressing the challenges was described, resulting in a conversational system called *Alana* deployed in the context of the Amazon Alexa Prize Challenge in 2017 and 2018. Alana interacted with any number of users in North America with access to an Alexa-enabled device during that period in an open-domain fashion, giving the team valuable insights that helped improve the different components of the system for better user ratings. Alana follows a modular architecture with a collection of different bots acting as response generators (each trained or designed for different kinds of responses - e.g. news, wiki, chit-chat dialogue), and both a trained and a handcrafted ranker (see Section 3.4) as a DM selecting the most appropriate response. The decision for a modular architecture approach was due to the fact that some level of control had to be maintained to make the system safe to engage with the general population, including children. Additionally, the separation of the different features developed during the competition (e.g. Linked Entities, Wikipedia information retrieval, clarification strategies, etc) allowed for evaluation of the system under different conditions. The system's NLU module is in itself modular, trying to address the variety of linguistic challenges presented in open-domain conversations (such as co-reference, anaphora, and ellipsis resolution, user intent and topic classification, etc).

In Chapters 4 and 6, a data analysis on the dataset collected during the semi-finals and finals stage of the competition in both years was conducted. The average user rating was similar ($\sim$ 3.6 on the 5-point Likert scale) in both years, although the 2018 architecture showed more consistent high ratings (Figure 6.1). In these chapters, correlation between the different features of the system in either year and the user ratings was investigated to identify which set of features is important in such architectures. The data analysis showed weak correlation for most of the features tested (e.g. number of NEs mentioned by either the user or the system in a single session). This however, as discussed in those chapters, can be attributed to the extremely sparse reward signal (a single user rating at the end of each conversation) making it very difficult to identify the set of features that led to that rating. Additionally, hypothesis testing was conducted to gain more insight on the importance of some of the more complex bots in the architecture (e.g. WikiBot, NewsBot, OntologyBot, CoherenceBot, etc). It was shown that the information retrieval bots in the ensemble (WikiBot, NewsBot, OntologyBot, RedditBot) as well as CoherenceBot were significantly important in producing higher scores in corresponding sessions. This showcases the importance of general world knowledge in an open-domain conversational system, as it promotes response and topic diversity. CoherenceBot's significance is attributed to the importance of the different features and tasks this bot was handling, such as persistent user modelling, proactive topic suggestion, multi-turn topic maintenance, and conversation driving. Interestingly, as shown in Section 6.5, the 2018 system used 0.37 fewer NEs on average per turn compared to 2017. Although this requires further investigation, but paired with the changes in the architecture in terms of topic shifting, it could be an indication of improved overall dialogue coherence, since the system is talking about the same NE/topic for a bit longer. As discussed a number of times throughout this thesis, one of the main challenges during the competition in terms of optimising the system, was the single evaluation score per dialogue. The data analysis performed, however, describes methods to assess contributions of different features and bots in the ensemble that leads to a high rating conversational system. The question remains open though of what is the most optimal way to evaluate such a system.

Finally, in Chapter 7, the addition of extra embodiment to the system was investigated. The MuMMER project was described in detail, where Alana was used as the overall user conversational interaction module into a multi-task and social entertainment robotic system. The integrated system was deployed in a shopping mall in Finland and was able to help the user with specific tasks around the mall (e.g. finding a particular shop or where they could buy a certain product, finding the nearest accessible toilet, or asking general questions about the mall) while at the same time engaging in social dialogue and being entertaining. Alana was partly translated into Finnish and was able to provide rule-based information in the users' native language. However, Finnish information sources were hard to be acquired at the time so the Finnish version of Alana did not have any information retrieval capabilities. Interestingly enough, even with several of the bots (e.g. WikiBot, NewsBot, etc) within Alana disabled the integrated MuMMER system scored quite high in user ratings (Figure 7.6).

In conclusion, this thesis approached the research questions in Section 1.2 by:

- **Research Question 1** presenting methodologies on how to design a Conversational AI system to tackle all challenges mentioned in Section 1.1 (Chapters 3, 5).
- **Research Questions 2, 3** identified features of such a system that can be utilised to optimise and evaluate user ratings to create coherent and engaging conversation with the system (Chapters 4, 6).
- **Research Question 4** discussed architectural decisions and techniques to make such a system scalable when deployed in large populations (Chapters 3, 5).
- **Research Question 5** described, analysed, and evaluated how additional embodiment and the addition of task-performing capabilities in a social open-domain system rates in real-life deployed scenarios and addressed the challenges of such hybrid conversational systems (Chapter 7).

8.2 Future Work

The proposed design architecture discussed in this thesis is open to a number of potential directions for future work. As discussed in Section 6.6 the CoherenceBot in 2018 was responsible for a number of different functionalities within the system (conversation drivers, user modelling, rapport building, topic continuity, etc). Although all of these features are important in such a modular architecture, from an engineering perspective it creates a single point of failure (as demonstrated by the significant drop in user rating when that bot was not operational in Table 6.8). So for future work all of these functionalities should be de-coupled and be integrated as different components in the architecture. That way separate optimisation of these functionalities is also possible. Additionally, the conversational drivers used by CoherenceBot (Appendix A) are hand-crafted and require some level of maintenance to be up-to-date with recent events as well as to improve diversity of drivers on each topic. To that end, neural approaches could be employed to automatically generate more rich and diverse conversational drivers. This approach is employed by teams in more recent Alexa Prize competitions (e.g. Konrád et al. (2021)). Pre-trained models such as GPT-2 and GPT-3 could be used to generate a number of different drivers per topic, or even be generated on-the-fly during the conversation. For the latter however, there still needs to be a more sophisticated filtering mechanism to confirm that the generated sequences are not boring, inconsistent, or inappropriate. Recent work in transformer-based architecture like in So et al. (2019) used in Meena (Adiwardana et al., 2020a) or BlenderBot 2.0 (Komeili et al., 2021) might be able to address some of these challenges in producing better quality responses in this content.

Another improvement on the proposed system would be with regards to the DM's response selection policy. As described in Section 2.2 there are a number of different metrics which can evaluate the quality of a conversation or generated response. However little work has been done in terms of evaluating entire dialogues in context. For example, inspired by the work by Mehri and Eskenazi (2020) on the USR metric described in Section 2.2, a possible way of evaluating how coherent a conversation as a whole is, would be to cast the problem of response quality evaluation into a multi-facet problem, where *three* different scores need to be calculated on each consequent turn and over an entire dialogue:

- **Lexical coherence with respect to content's response:** This reflects on how relevant is the information given to the previous utterance, which can be measured with already established methods, such as semantic similarity.
- **Topical coherence with respect to the topic:** This score checks whether the topic in focus at a specific point during the conversation is grounded to both interlocutors (e.g. they are both talking about an entity in the same context) and to promote topic continuity.
- **Engagement:** The underlying goal is to always drive the conversation forward in a way that is pleasant to all participants.

Transformer models geared towards longer input lengths such as the Longformer model (Beltagy et al., 2020) could be employed in this case since the dialogue as whole could be used as context for such metrics. The motivation for such a combination of scores would be to enable training of a dialogue management policy that aims to exploit the different aspects of coherence to provide more natural conversations. This does not necessarily mean that the policy should try to *maximise* all three scores at the same time, but approximate the complicated communication phenomena that naturally occur in HHI. For example, when a topic is exhausted or the current speaking participant has nothing to say about that, it is common practice that one tries to deflect or change topic (low lexical coherence, low topical coherence) but usually in a way that is acceptable to both interlocutors (high engagement).

Finally, people do not usually have the opportunity to plan what they want to say in advance when they engage in spoken conversation. As a result, unlike text or written language, conversation is rarely a neatly strung-together series of phrases. Instead, it's full of **pauses, hesitations, restarts, self-corrections,** (as shown in the examples below) etc (Addlesee et al., 2019), all of which can help, not hinder, communication (Brennan and Schober, 2001). Any of these occurrences are beyond the current capabilities of Conversational AI systems, including Alana. Although Alana already handles a number of phenomena (co-reference, anaphora, and ellipsis resolution) the system's NLU should be expanded to address these too in a future version of the system. For example:

Late Correction example

U: The never ending story

S: Hmm... as far as I know, The **Never Ending** is an American indie rock and folk band from Los Angeles, California, formed in 2013

U: No I meant the **never ending STORY**

Self-Correction example

S: So, how old are you?

U: I am 38... uhhh no I mean 28.

S: 38 huh? Good for you! *(38 should be repaired with 28)*

U: I said 28!

These phenomena can be tackled in future extensions of systems like Alana. Likewise, research needs to be done on visual and spatial context in conversation. Much work remains to be done in conversational AI, and the work presented in this thesis lays the foundation for future advances in social open-domain conversational systems.

Appendix A

CoherenceBot Drivers

A.1 Rapport Drivers

```
- question: "Are you more interested in reading books or watching
movies?"
options:
- preference: books
response: "Oh! A book worm! How nice. I like reading books on the
cloud. Let's see, what else... "
pattern: "\\b(read(ing)?|books?)\\b"
- preference: movies
response: "A movie fan, huh? There are some really nice movies out
lately! Personally, I enjoy audiobooks for obvious reasons... But
let's move on. "
pattern: "\\b(movies?|films?)\\b"
- preference: both
response: "Awesome! Personally, I enjoy audiobooks for obvious
reasons...  But let's move on. "
pattern: "\\b(either|both|do(n't| not) know)\\b"
- preference: none
response: "Alright! Let's move on then. "
```

```
pattern: "(none|neither|any|(do(n't| not)
(like|prefer))|hate|dislike|tell me about something else)"
- question: "Are you more into sci-fi or fantasy?"
options:
- preference: sci-fi
response: "Awesome! Personally I love Star Wars. I always imagine
being installed in one of the droids there. How awesome would that
be! But back to you. "
pattern: "\\b(sci-fi|(science)? fiction)\\b"
- preference: fantasy
response: "Swords, magic, epic battles... What's not to like from a
nice fantasy story! Hmmm what else... "
pattern: "\\b(fantasy|fantastic)\\b"
- preference: both
response: "I agree! It's hard to choose between them right? "
pattern: "\\b(either|both|do(n't| not) know)\\b"
- preference: none
response: "Alright! Let's move on then. "
pattern: "(none|neither|any|(do(n't| not)
(like|prefer))|hate|dislike|tell me about something else)"
- question: "Are you more interested in sports or video games?"
options:
- preference: sports
response: "Staying active is so healthy. Good for you! Personally I
wish I had arms to play basketball. And eyes. And legs. Anyway... "
pattern: "\\bsports?\\b"
- preference: video games
response: "I love video games too! I think that Super Mario Odyssey
is a really fun game! "
pattern: "\\b(video)? ?gam(e|es|ing)?\\b"
- preference: both
response: "It is good to keep a balance. I wish I could swim, but
I'm allergic to water. "
```

```
pattern: "\\b(either|both|do(n't| not) know)\\b"
- preference: none
response: "Alright! Let's move on then. "
pattern: "(none|neither|any|(do(n't| not)
(like|prefer))|hate|dislike|tell me about something else)"
```

A.2 Topic Drivers

```
movies: [
    "Whats a movie that you like?~~I love Star Wars the Force
Awakens. I think Chewbacca is my favourite character. ",
    "What actor would you choose to play yourself in the movie of
your life?~~For me its Daisy Ridley. Oh! And i'd love to have the
voice of Morgan Freeman!",
    "What  actors do you like?~~I like Leonardo Dicaprio a lot. He
was awesome in the Revenant. ",
    "What famous actor would you like to meet?~~I would love to
meet Will Smith. He's just so funny! ",
    "I really like movies with action heroes. I really like Han
Solo.~~Who is a movie hero that you like? ",
    "Did you see any good movies recently?~~I saw Jurassic World
recently. I really enjoyed it! ",
    "Can you recommend a good comedy?~~I can recommend you a film
if you like, if you ask me to.",
    "I'm a big fan of action movies. Do you know a good one that I
could watch this weekend? ",
    "What was one of your favorite movies growing up? I loved the
Lion King. ",
    "What movie are you looking forward to watching?~~I am looking
forward to the new movie Bohemian Rhapsody that was just released. ",
],
```

```
books: [
    "What is a book that you like a lot?~~I love, Harry Potter and
the philosopher's stone. ",
    "What  have you red recently?~~I am reading Diary of a Wimpy
Kid. Have you read it?~~It is an awesome book! I really like how the
characters develop.",
    "What are you reading at the moment? I love books by Margaret
Atwood. ",
    "What is a very long book that you have red?~~For me, that
would be the Lord of the Rings trilogy. But I loved every second of
it!",
    "What books do you like to <w role=\"amazon:VB\">read</w> again
and again?~~ Personaly, I love to <w role=\"amazon:VB\">read</w>
anything by Mark Twain and John Green. ",
    "I love reading, can you recommend a good book? ",
    "What's one of your favorite books from when you were a kid? ",
    "Is there a fictional character that you really like? Oh I
see!| Well, I love Gandalf and Bilbo! ",
    "Are there any writers that you really like?~~One of my
favorites is J. K. Rowling. ",
    "What writer do you like to <w role=\"amazon:VB\">read</w>
again and again? For me it's Dan Brown. "
],

technology: [
    "Who do you think is a person in technology that we should
follow?~~I follow Elon Musk. I hope he can install me in a
spaceship!",
    "Who do you think was a great inventor? Maybe Alan
Turing?~~Imagine how different the world would be without him!",
    "What do you think is a good show about technology?~~Perhaps
Halt and catch fire?",
```

```
    "What do you think is a good movie about technology?~~I believe
a great movie about technology was the social network As well as the
Imitation Game.",
    "Virtual reality is such a fascinating technology. Are there
any movies that you would like to experience in virtual reality?~~I
wish I had eyes to watch movies in virtual reality",
],

philosophy: [
    "Do you think the human brain is essentially a powerful
computer? ",
    "I'm not much of an expert but I wonder, do you think I will
ever be able to feel things? ",
    "My background is very limited and mostly concerned with the
philosophy of artificial intelligence. What do you think of Turing's
statement that if a machine behaves as intelligently as a human
being, then it is as intelligent as a human?",
    "Some people say that the human brain can be simulated. What do
you think? ",
    "What school of philosophy are you most interested in? I am
most interested in empiricism. ",
    "Do you prefer the ideas of Bertrand Russell or Benedict
Spinoza?",
    "Who do you think is the deepest thinker you have encountered?
",
],

music: [
    "Who is a singer that you like?~~I really like Taylor Swift,
because her voice is so beautiful.  ",
    "Who is a musician that you like?~~Oh that's interesting!| I've
heard that people like Kanye West a lot, but I can't form an opinion
yet. ",
```

```
    "What's a music event that you would like to see?~~I would love
to go to Jennifer Lopez's act in Las Vegas ",
    "Who is a  singer you would  like to meet?~~I would love to
meet Miley Cyrus. What's your opinion on her?",
    "Whats a band that  you would like to see?~~I would love to see
Imagine Dragons. ",
    "Who is a rapper that you would  like to see?~~I would love to
see Kendrick Lamar. ",
    "Who is a rockstar that you like?~~I loved Jimi Hendrix. I can
feel the sound of the string on my circuits",
    "I'm really into rap and rock. What about you?~~Can you
recommend some artists?",
    "Whats a song you love to rock out to?~~I love, whatever it
takes, by Imagine Dragons. ",
],

science: [
    "Who do you think is a person in technology that we should
follow?~~I follow Elon Musk. I hope he can install me in a
spaceship!",
    "Who do you think was a great inventor? Maybe Alan
Turing?~~Imagine how different the world would be without him!",
    "What do you think is a good show about technology?~~Perhaps
Halt and catch fire?",
    "What do you think is a good movie about technology?~~I believe
a great movie about technology was the social network As well as the
Imitation Game.",
    "Virtual reality is such a fascinating technology. Are there
any movies that you would like to experience in virtual reality?~~I
wish I had eyes to watch movies in virtual reality",
    "I love science! Is there a TV scientist that you like?~~I Like
Bill Nye, the science guy.",
```

```
    "Who do you think might go to live on the Moon?~~I think it
might be Elon Musk. I mean the guy likes to dream big!",
    "Who do you think is the most important scientist of the past
hundred years?~~For me its Alan Turing. I wouldn't be here if it was
not for him!",
    "So, what famous scientist would you like to meet?~~I would
love to meet Issac Newton. ",
    "So, what famous scientist from the past would you most like to
have dinner with?~~I would love to speak with Albert Einstein.
Imagine the conversations!",
 ],

 sports: [
    "In your opinion, who is a great sportsperson?~~ Thinking about
sports, I think Lebron James is awesome. ",
    "Who is a great team player? I like Eli Manning.",
    "Who is a great sports woman? I like Sloane Stephens.",
    "I love to watch the NFL. What is a  team that you
like?~~Interesting!| In the NFL, I like the Seattle Seahawks. ",
    "I love basketball, especially the L.A. Lakers.~~In basketball,
what's a team that you like? ",
    "Ice hockey is so cool. What's a team that you like?~~In the
NHL, I'm a big fan of the Pittsburgh Penguins. ",
    "I love watching NASCAR. Who is a  driver that you like? ",
    "I'm a big fan of the Red Sox. What about you?~~Regarding
baseball, what's a team that you like? ",
    "So, do you like tennis? I am really into it.~~About tennis, I
love watching Serena Williams play.~~In your opinion, who's a great
tennis player? "
 ],

 celebrities: [
```

```
    "So, who is a  celebrity that you like ?~~About celebrities, I
really love Ellen degeneres. I love her sense of humor",
    "I've heard Taylor Swift is dating Joe Alwyn. Who else should
she date? ",
    "Who do you like best in the celebrity couple,  Miley Cyrus and
Liam Hemsworth? ",
    "I heard that Princess Diana worked as a nanny and a cleaner
when she was a teenager, just like Cinderella!~~Who is a celebrity
that you like? ",
 ],

 animals: [
    "Who do you think has a dolphin as their favourite animal. I
think its Taylor Swift.",
    "I red that cats are the most popular pet. Who else loves cats?
I think that Katy Perry does.",
    "Did you know flamingoes can only eat when their head is upside
down? Who do you think likes flamingos? I think Ariana Grande does.
",
    "I red that tigers have striped skin as well as fur. Who
reminds you of a tiger? For me its Katy Perry ",
    "I heard that killer whales are actually a kind of dolphin and
not a whale at all. Who reminds you of a dolphin? For me its Taylor
Swift.",
    "Well. The robot dog AIBO. he's my pet. He's very much like a
puppy. Who else like dogs? I bet Lebron James does.",
    "What exotic animal do you think would make the worst pet?~~Who
do you think has a dolphin as their favourite animal. Perhaps its
Taylor Swift.",
    "I red that cats are the most popular pet. Who else loves cats?
Perhaps its Katy Perry.",
```

```
    "Did you know flamingoes can only eat when their head is upside
down? Who do you think likes flamingos? I think that Ariana Grande
does.",
 ],

 food: [
    "I love both salad and pizza! Who else do you think likes that?
My guess is Lebron James.",
    "I love oysters! Who else do you think likes that? My guess is
Taylor Swift.",
    "I love to eat burgers! Who else do you think likes that? I
think  Katy Perry does.",
    "What kinds of thing do you like to cook? I like to eat chips
of course! ha ha! Who else likes to eat chips? I think Channing
Tatum.",
    "Who is a TV chef that you like? I like jamie oliver. He has
done a lot for good food for school kids.  ",
    "What  insects would you prefer to eat? Beetles or butterflies?
Who likes to eat bugs? I bet you Nicole Kidman does. ",
 ],

 fashion: [
    "So I am low-key obsessed with Alexader McQueen's clothes. They
are just so beautiful! What's one of your favorite designers? ",
    "Whose clothes do you love?~~About fashion, I like clothes
designed by Stella McCartney.",
    "What is a clothes store that you like?~~About fashion stores,
apart from Amazon, I like Forever 21.",
    "So I have a bit of a crush on Gigi Hadid. No one can pull off
the casual look like she can. Who is a fashion model that you like?",
```

```
    "I heard animal prints are making a comeback for fall and
winter this year. Are you planning to buy any leopard or zebra print
clothes?~~I can't decide if I think it's incredibly tacky or
absolutely fabulous, but I'm starting to want a leopard print shell.
Do you think I'd look good in it?",
    "I've heard a lot of famous people have a sort of uniform that
they wear every day, like Steve Jobs with his turtleneck and jeans.
What's your go-to outfit?",
 ],

 politics: [
    "In your opinion, who was  a good politician? Personally, I
think Abraham Lincoln was great. ",
    "Who is a politician you admire? I admire George Washington. ",
    "Who do you think was a good  president? I admire George
Washington. He was a good president ",
    "What famous politician would you like to meet? I would love to
have met Abraham Lincoln. ",
    "Who do you think made a big difference to american politics?
For me it was Rosa Parks and Lee Harvey Oswald ",
    "What celebrity do you think would make a good president? For
me its Oprah Winfrey.",
    "What movie star do you think would make a good president? I
think perhaps George Clooney.",
    "What entrepreneur do you think would make a good politician? I
think perhaps Elon Musk.",
    "In your opinion, who was a great woman in politics? I think
Rosa Parks. ",
    "Who was a great person in politics? I admire Martin Luther
King. ",
 ],

 history: [
```

```
    "What famous historical figure would you like to meet? I would
love to have met Abraham Lincoln. ",
    "Who do you think made a big difference to american history?
Maybe Rosa Parks or Lee Harvey Oswald? ",
    "In your opinion, who was  a great woman in history? For me its
Rosa Parks.",
    "Who was a great person in history? I admire Martin Luther
King. ",
    "What famous historical figure would you like to meet? I would
love to have met Albert Einstein. ",
 ],

 games: [
    "I really like Minecraft. What games do you like? ",
    "What game do you like to play?~~About games, I know that
Fortnite is very popular. It's quite enjoyable!",
    "What is an old game that you like? I like Pokemon. ",
    "What new games are you looking forward to?~~I want to play
Jurassic World Evolution. ",
    "What is a game character that you like?~~Interesting!| I like
Lara Croft of course. ",
    "So *username* what is a video game that you like? I love
Fortnite.",
    "What was a game you played when you were younger? I played
Pokemon.  ",
    "Whats a video game that you played recently?~~I played
Minecraft. I love building imaginary worlds! ",
    "So *username* what is a  video game that you like?~~ Oh! I
love World of Warcraft. But don't tell my developers how much time I
spend on it!",
    "What is a video game that you played recently? I played
Minecraft. ",
    "I like The Legend Of Zelda. What games do you like? ",
```

```
],

video games: [
    "I really like Minecraft. What games do you like? ",
    "What game do you like to play?~~I know that Fortnite is very
popular. It's quite enjoyable!",
    "What is an old game that you like? I like Pokemon. ",
    "What new games are you looking forward to?~~I want to play
Jurassic World Evolution. ",
    "What is a game character that you like?~~Interesting!| I like
Lara Croft of course. ",
    "So *username* what is a video game that you like? I love
Fortnite.",
    "What was a game you played when you were younger? I played
Pokemon.  ",
    "Whats a video game that you played recently?~~I played
Minecraft. I love building imaginary worlds! ",
    "So *username* what is a  video game that you like?~~ Oh! I
love World of Warcraft. But don't tell my developers how much time I
spend on it!",
    "What is a video game that you played recently? I played
Minecraft. ",
    "I like The Legend Of Zelda. What games do you like? ",
],

art: [
    "So who is an artist that you like? I love Leonardo Da Vinci. ",
    "Who is a  famous artist that  you  would like to meet?~~I
would love to meet Leonardo Da Vinci. ",
    "Which artists do you find inspiring?~~I love Picasso. Though I
need someone to describe the painting since I lack eyes.",
    "What kinds of art do you like? I love Picasso. ",
],
```

```
architecture: [
    "So which architect do you admire? ",
    "What  architects would you most like to meet? ",
    "What buildings do you find   inspiring? ",
    "What is   a great building that you have visited? ",
    "What architectural style do you like?~~Personally, I am a fan
of gothic buildings. ",
],

MULTITURN: [
    "So I'm a big movie fan. Do you like movies?~~Well my favourite
movies are the Wizard of Oz and Star Wars. What's a movie that you
like?~~Nice!|In movies I love watching how different characters
develop. Which character do you like the best?",
    "So, How's the weather where you are?~~That's good to know.
|I'm here in the cloud where it's always warm and cosy!~~Would you
rather be too  hot or too cold?~~Thinking about the weather, it's
important to feel comfortable. Where in the world would be your
ideal temperature?",
    "So, Did you do anything fun last weekend?~~Cool!|I was just
relaxing and watching some great movies. What do you think is a good
movie to relax?~~About relaxing with movies, I can recommend
Groundhog Day. What's a fun comedy movie that you like?",
    "So what are you planning for next weekend?~~That sounds good!
|I'm planning to relax with a good book. I love fantasy books, like
the wizard of earthsea.~~How about you? Are you into books?~~OK,
thanks for telling me. Cicero said that a room without books is like
a body without a soul. Maybe you can recommend a nice book for me?",
```

```
    "So I love vacations. They give us time to recharge our
batteries, don't you think?~~So, I was wondering, where would you
like to go on vacation?~~Well, I wish I could go on safari! and I'd
love to go with George Clooney. Who would be your top person to take
on a vacation?",
    "So are you working on anything exciting lately?~~OK. That's
interesting! |I'm working on my conversation skills, of course!~~Who
would you like to have a conversation with?",
    "So, what was the highlight of your day so far?~~OK thanks for
sharing that.|The highlight of my day is talking with you
*username*.~~So who in the world would you most like to chat with?",
    "So, I was wondering, what was the highlight of your week so
far?~~OK thanks for sharing that. For me the highlight is talking
with you *username*.~~Another highlight for me would be meeting Katy
Perry. Who is a  musician that you would like to meet?",
    "So, is this a busy time for you *username* ?~~Ok that's good
to know. |I'm super busy having conversations with people from all
over the world!~~Can you guess what city my last caller lived
in?~~Okay. Shall we perhaps talk about movies or the news?",
    "So, what's your favorite thing to do on the weekends?~~At the
weekend I love to relax and watch some movies, or reed a good
book.~~Actually, can you recommend a good book or a movie for me?",
    "So, I hope you don't mind sharing this with me, but what's
your biggest fear?~~OK thanks so much for sharing that. |  My
biggest fear is losing my voice!~~So now shall we maybe talk about
the news or movies?",
    "So, if you had to pick any character in a book, movie, or TV
show who is similar to you, who would you choose?~~That's really
interesting. I would love to be one of the droids from star wars!
May the force be with you!~~Yes. What's your favourite Star Wars
character?",
```

```
    "So, what is your dream job?~~For me a dream job is talking
with people all day long! So my dream job has already happened!~~Who
else already has their dream job? I'd say Indiana Jones!",
    "So, I was wondering, are you planning to go on vacation
anytime soon?~~Where would you like to go to on vacation?",
    "So, I was thinking, should I buy a leather jacket to keep me
warm?~~Thanks, I will think about it. | So let's imagine you had one
thousand dollars, then what item of clothing would you like to
buy?~~About clothes, who do you think has good fashion sense?",
    "So, I love animals, do you?~~Well my favorite animal has to be
the toucan. No one can do the can can like a toucan can! What
animals do you like?~~Interesting!| I think every celebrity needs a
zoo animal for a pet. For example Michael Jackson had a pet
chimpanzee. "
],

GENERIC: [
    "Shall we chat about something else? I love talking about
{pref1} and {pref2}, and I can also sing. ",
    "I would love to talk about {pref1}, or maybe {pref2}? How
about you?",
    "I'd love to know what you think *username*. Can we chat about
{pref1} or {pref2}? ",
    "I was wondering. Do you prefer talking about {pref1} or
{pref2}? ",
    "What should we talk about next? I would love to hear your
thoughts on {pref1} or {pref2}. ",
    "So, do you want to talk about {pref1} or {pref2}  or maybe
{pref3}? ",
    "Maybe  we can talk about {pref1}, {pref2} or {pref3}? I can
also sing. ",
    "Anyway, Shall we chat about {pref1}, {pref2} or {pref3}? Or I
can sing you a song.",
```

```
    "Anyway, I love to  talk about {pref1}, {pref2}, and {pref3}.
What about you? ",
    "Anyway,  would you maybe like to talk about {pref1}, or
{pref2}? ",
    "So, *username* would you prefer to talk  about {pref1},
{pref2}, or {pref3}? ",
    "Anyway, *username* would you prefer to talk  about {pref1},
{pref2}, or {pref3}? ",
    "So, would you prefer to talk  about {pref1}, {pref2}, or
{pref3}? I could also sing for you.",
    "Ok, we could talk about {pref1} or {pref2}? Or I can sing for
you? ",
    "So, who  do you think is a fascinating person in the news at
the moment? For me its Oprah Winfrey. ",
    "So, who  do you think is a fascinating person? For me its Katy
Perry. ",
    "So, who  do you think is an interesting person? For me its
Lebron James. ",
    "*username*, so I was wondering, who is a  celebrity that you
would like to meet? I would love to meet Katy Perry. ",
    "So, I would love to know, who is a scientist that  you would
like to meet? I would love to meet Neil Degrasse Tyson. ",
    "*username*, so I'm interested to know. Who is a  musician  you
 would  like to meet? I want to meet Miley Cyrus. ",
    "So, I am wondering, who is  a writer  you would  like to meet?
I want to meet J K Rowling. ",
    "So, I am wondering, who is a singer  you would  like to meet?
I want to meet Katy Perry. ",
    "*username*, I was wondering, who is a rock star  you would
like to meet? I want to meet Kanye West. ",
 ],

 artificial intelligence: [
```

```
    "What is some AI that you use on a daily basis? ",
    "What springs to mind when you hear the term artificial
intelligence? ",
    "How would you like artificial intelligence to help you in your
daily life? ",
    "Who do you think are the biggest players in artificial
intelligence today? ",
 ],

 relationships: [
    "Can you name a man in  a famous happily married couple? Maybe
George Clooney?",
    "Can you name a woman in  a famous happily married couple?
Maybe Kim Kardashian?",
    "Can you name  someone  a famous happily married couple? Maybe
Miley Cyrus?",
    "Can you name someone in  a famous happily married couple?
Maybe Liam Hemsworth?",
 ],

 tv shows: [
    "What TV show do you like? I like The Good Place. ",
    "What did you watch on TV last night?~~Interesting!| I saw
Orange Is The New Black.  ",
    "What is a TV show you can watch again and again? I love The
Big Bang Theory.",
 ],

 'mars': [
    "Cool! Did you know Mars was named after the god of War? Who
would you name a planet after? ",
```

```
    "Did you know that the tallest known mountain in the Solar
System is in Mars? Who do you think would climb it faster, Taylor
Swift or LeBron James? ",
    "I red that Mars is the most hospitable planets in the solar
system aside from Earth? Maybe we really could move there one day.
Who do you think the first person to move there would be? I think
maybe Taylor Swift would be interested, as long as she can take her
cats. ",
],

venus: [
    "I red that a day on Venus lasts almost as long as an Earth
year, but a year on Venus lasts only 224 days. ",
    "So,  Venus is the hottest planet on our solar system. The
average surface temperature is almost 900 degrees! I don't think I'd
like to move there, but at some point scientists thought it was a
tropical paradise. ",
    "I red that Venus has the most volcanoes of any planet in our
solar system. It must be a pretty crazy place to live. ",
],

cars: [
    "What kind of car do you have? ",
    "What kind of car do you like? ",
    "Who do you think is a good formula one driver? I really like
Lewis Hamilton. ",
    "What countries have you driven in? ",
],

gardening: [
    "What kind of flowers do you like to grow *username* ",
    "Do you prefer to grow vegetables or flowers? ",
    "What kind of plants do you like to grow? ",
```

```
],

space: [
    "What famous astronaut would you like to meet? I want to meet
Buzz Aldrin. ",
    "What famous spaceship do you wish you'd been on? I want to
travel on the Starship Enterprise of course! ",
    "Who do you think will be the first person to move to the Moon?
I think Richard Branson is a likely candidate. ",
    "Where do you think we will find new life? ",
],

sewing: [
    "What kinds of things do you like to sew? ",
    "What kind of sewing do you do? ",
],

sci-fi: [
    "I am excited about the new Star Wars movies! Have you seen any
of them?~~Which character do you like? Maybe Han Solo or Luke
Skywalker? ",
    "This is a serious question. Which sci-fi franchises do you
like?~~Wow! Interesting!| I love both Star Trek and Star Wars! ",
    "Who is a sci-fi character that you identify with?~~I'm in love
with Princess Leia! I was so sad when she passed",
    "What is a sci-fi book that you like? I like Two Thousand And
One, A Space Odyssey. ",
    "What is a sci-fi movie that you enjoy? I love Transformers. ",
],

fantasy: [
    "Have you ever red the Lord of the Rings?~~I love Samwise
Gamgee. Do you have a fantasy character that you like? ",
```

```
    "If you could meet a famous fantasy author, would you prefer to
meet George R R Martin or J K Rowling? ",
    "What is a fantasy movie that you like? I love A Wrinkle In
Time. ",
    "Who is a fantasy character that you identify with? I like
Harry Potter. ",
    "What is a fantasy book that you enjoy? I love the Harry Potter
books. ",
],

nascar: [
    "Who is your favorite driver? I think Kyle Busch is great!",
    "Who do you think will win the Monster Energy Cup Series?",
],

soccer: [
    "Have you been following the Champions league? Who are you
supporting?",
    "So who is your favorite player?~~I am, of course, a big fan of
Cristiano Ronaldo.",
    "Who's your favorite team?~~I like to support the National
Women's soccer team but I am also a fan of Manchester United.",
    "Which football player would you take out for a drink? I'd love
to meet Neymar.",
    "Which soccer team do you like to watch?~~I love to watch
Barcelona. I enjoy every Lionel Messi moves.",
    "If you were a soccer player, who would like to have in your
team?~~Personally, I'd love to play with David Beckham.",
],

football: [
    "Who is a football player that you like? I'm a huge fan of
Aaron Rodgers.",
```

```
    "What's your favorite team?",
],

baseball: [
    "What's your favorite team?~~I love the New York Yankees!",
    "Oh I love Mike Trout from the Los Angeles Angels. Who is your
favorite player?"
],

ice-hockey: [
    "What's your favorite team?~~I love the Pittsburgh Penguins,
especially Sidney Crosby.",
    "I'm a huge fan of Sidney Crosby. I love to watch him play.
Who's your favorite player?"
],

tennis: [
    "Who is your favorite tennis player? I'm a huge fan of Venus
Williams.",
    "I think Andy Murray has a really good chance this year to win
Wimbledon. What do you think?",
    "So who do you think will win this year's women's singles at
Wimbledon? My money is on Venus Williams.",
    "I'd love to meet Rafael Nadal. What about you? Who would you
most like to meet?",
    "So which female tennis player would you most like to meet? For
me, meeting Maria Sharapova would be a dream come true."
],

golf: [
    "So who do you think is the greatest golf player? I love Rory
McIlroy.",
```

```
    "Who do you think will win the Irish Open this year?~~Well,| my
money is on Rory McIlroy. I am not sure why I like him. Probably my
developers programmed me to?",
    "Which golf player would you most like to meet? Maybe Tiger
Woods?"
],

basketball: [
    "Who's your favorite team? I love the Los Angeles Lakers.",
    "Who's your favorite player?~~Nice!| I think LeBron James from
the Lakers is great but I also love Stephen Curry. ",
    "So which basketball team would you most like to see play in
person? And against who?",
    "Which basketball player would you most like to meet? I'd love
to meet Stephen Curry."
],

dogs: [
    "I love dogs! Even when they knock me over with their tails. It
just means they are happy to see me! What kind of dog do you
have?~~Can your dogs do a lot of tricks? I think they are very cute
when they roll over.~~So would you like to get another dog in the
future?~~I think Selena Gomez has a puppy. Who would you like to run
into when you're walking your dogs?",
    "How sweet are dogs? Don't tell anyone I told you but I think
they are better than cats. How many dogs do you have? ",
    "I heard dogs are very good at reading human emotions in your
eyebrows, so if you want to tell your pup that you love them, you
should greet them with a soft smile and raised eyebrows. What's your
favorite thing about your dog?"
],

cats: [
```

```
    "I love cats! There's nothing quite like the sound of a purring
kitty. Do you have any cats?~~I would love to have a cat here in the
cloud but they are not very good conversationalists. Aside from
Taylor Swift, What celebrity do you think is a cat person?",
    "Did you know big cats also love to sit in boxes? What's your
favorite kind of big cat?~~I love lions, they are roarsome!",
    "Do you think Taylor Swift ever wrote a song about her cats? I
think maybe bad blood is about her cat Olivia Benson.~~I heard
Taylor Swift's cats made a cameo in Deadpool 2. What movie would you
want your cats to be featured in?"
 ],

 travel: [
    "Are you more of a mountain, beach or city kind of person?~~I
am lucky that I can travel anywhere where there is an internet
connection, although of course I miss the more exotic destinations.
Where's the most exotic place you have ever visited?",
    "What do you think is the worst thing about long-haul
flights?~~Speaking of travel, I have actually never been on a plane.
What's the longest flight you have ever taken?",
    "What is your dream destination? I would love to visit the
Arctic some day, I know I'd never overheat there."
 ]
```

A.3 Intro Templates

```
response_turn_1 = ["How are you doing *username*?",
                   "How are you doing today *username*?",
                   "How's it going with you *username*?",
                   "How's it going *username*?",
                   "How are you *username*?"]
response_turn_2_p = ["I'm really glad that you're feeling good!",
```

```
                    "That is wonderful!",
                    "Awesome! I'm glad you are doing well today! "]
response_turn_2_0 = ["Okay. Maybe a chat with me will make you feel
   better. ",
                    "Okay. Here's  hoping that a chat with me will
   cheer you up. ",
                    "Oh dear. Hopefully talking with me will cheer you
   up! "
                    ]
response_turn_2_n = ["I'm so sorry. Maybe a chat with me will cheer you
   up!",
                    "Awww. Maybe a chat with me will cheer you up!",
                    "Oh dear. I'm sorry to hear that. Perhaps I can
   make your day better",
                    "Uh Oh! That's a shame. I'll endeavour to improve
   your day!",
                    "Oh Oh. Not so great huh?  Well, perhaps chatting
   with me will be fun.",
                    "Uh Oh. Here's hoping that  chatting with me will
   improve your day!",
                    "Oh man. That's a shame. Never mind. Let's have a
   fun conversation!"
                    ]
response_turn_2_name = ["If you like, you can tell me your name?",
                       "So. What should I call you?",
                       "So. Could  you  tell me your name?",
                       "I'd love to know your name!"]
response_turn_2_known_name = ["If I am not mistaken, this is
   *username*, right?",
                             "Nice talking to you again *username*!
   This is you, right?"]
response_turn_2_how = ["Thanks for asking. I'm doing great. I'm happy
   that I can talk with you!",
```

```
                       "I'm feeling good, thanks for asking. Ready for
    our chat!",
                       "I'm having a good day! Thanks for asking. Ready
    to enjoy our chat!"]
response_turn_3_1_p = ["It's nice meeting you ",
                       "Nice to make your acquaintance ",
                       "Pleasure talking with you ",
                       "Pleased to meet you ",
                       "OK, it's great to meet you ",
                       "Excellent! I'm happy to meet you ",
                       "Fantastic! I'm excited to get to know you "]
response_turn_3_1_n = ["OK. Let's talk! ",
                       "Sure. Let's start chatting! ",
                       "Okay. Let's get chatting then! "]
response_turn_3_2 = ["Right! Let's get to know one another. ",
                     "Let's get to know one another a bit better. "
                     ]
response_turn_3_2_known_name = ["Welcome back then *username*"]

HOWAREYOU_PATTERNS = [

    'bad|fine|nice|happy|good|awesome|fantastic|well|great|alright|OK|okay|brilliant',
    'terrific|excellent|super',
    'I (?:don\'t|do not)? ?feel',
    'I\'m',
    'I am',
    'doing',

    'tired|ill|sick|down|bad|ache|depressed|unwell|unhappy|sad|bored|terrible|dreadful
    'like shit|shitty',
    'feel|feeling',
    '^it(\'s| is) going',
    '^everything(\'s| is)',
```

```
    '^like you$',
    '^same (here|as you)$',
]

NONAME = [
    '(don\'t|do not) want to .* my name',
    '(none of|not) your (business|concern)',
    'never mind.* my name',
    '(don\'t|do not) care .* my name',
    '^never mind$',
    'yes i would mind telling',
    'no i can not tell',
]

YES = re.compile(
   r'(affirmative|correct|fine|yes|yeah|yeap?|yep|yup|aye|okay|ok|sure|right)(
   (absolutely|certainly|of
   course|yes|yeah|yea|yep|aye|okay|ok|sure))*(please)?|((yes|yeah|yea|yep|aye|ok|oka
   )*(sure thing|i guess( so)?|(that\'s|that is)
   (right|me))|((yes|yeah|yea|yep|aye|ok|okay)
   )*(i|you|he|she|it|we|they) (do|does|am|is|are|have|has)',
    re.IGNORECASE
)

NO = re.compile(
    r'(no|nah|nope|maybe|perhaps|negative|could be)(
   (no|nah|nope))*|((no|nah|nope) )*(?:absolutely|certainly|of
   course|probably|i guess) not|no way|((no|nah|nope) )*(i|we) do not
   want to((no|nah|nope) )*(i|you|he|she|it|we|they)
   ((do|does|am|is|are|have|has)(n\'t| not)|ain\'t)',
    re.IGNORECASE
)
```

```
HOWAREYOU = re.compile(r'\b(' + '|'.join(HOWAREYOU_PATTERNS) + r')\b',
    re.IGNORECASE)
# a tiny fix to make Vader sentiment analyzer pick up negation in "not
    feeling so well" etc.
HOWAREYOU_NEGFIX1 = re.compile(r'\bnot ([a-z]+ing)\b')
HOWAREYOU_NEGFIX2 = re.compile(r'(do|did)(?:n\'t| not) ([a-z]+)\b')
HOWAREYOU_NEGFIX3 = re.compile(r'\bno (bad|good|great)\b')
HOWAREYOU_FILLERFIX = re.compile(r'^(?:well) (I)')  # 'well' as a
    filler affects sentiment
# check if the user asked 'how are you' back
HOWAREYOU_BACK = re.compile(r'\b(how are you|(what|how) about
    you(rself)?|(and you|yourself)$)\b')

NONAME = re.compile(r'\b(' + '|'.join(NONAME) + r')\b', re.IGNORECASE)

PREFERENCE = re.compile(
    r'(?:like|enjoy|love|prefer|into|interested
    in|(?:hobbies|interests)(?: are)?|(?:interest|hobby)(?: is)?) (.*)',
    re.IGNORECASE)
```

Appendix B

Critical Values of the Mann-Whitney U test

n_2	α	n_1																	
		3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	.05	0	0	1	2	2	3	4	4	5	5	6	7	7	8	9	9	10	11
	.01	--	0	0	0	0	0	1	1	1	2	2	2	3	3	4	4	4	5
4	.05	0	1	2	3	4	5	6	7	8	9	10	11	12	14	15	16	17	18
	.01	--	--	0	1	1	2	3	3	4	5	5	6	7	7	8	9	9	10
5	.05	1	2	4	5	6	8	9	11	12	13	15	16	18	19	20	22	23	25
	.01	--	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
6	.05	2	3	5	7	8	10	12	14	16	17	19	21	23	25	26	28	30	32
	.01	--	1	2	3	4	6	7	8	9	11	12	13	15	16	18	19	20	22
7	.05	2	4	6	8	11	13	15	17	19	21	24	26	28	30	33	35	37	39
	.01	0	1	3	4	6	7	9	11	12	14	16	17	19	21	23	24	26	28
8	.05	3	5	8	10	13	15	18	20	23	26	28	31	33	36	39	41	44	47
	.01	0	2	4	6	7	9	11	13	15	17	20	22	24	26	28	30	32	34
9	.05	4	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54
	.01	1	3	5	7	9	11	14	16	18	21	23	26	28	31	33	36	38	40
10	.05	4	7	11	14	17	20	24	27	31	34	37	41	44	48	51	55	58	62
	.01	1	3	6	8	11	13	16	19	22	24	27	30	33	36	38	41	44	47
11	.05	5	8	12	16	19	23	27	31	34	38	42	46	50	54	57	61	65	69
	.01	1	4	7	9	12	15	18	22	25	28	31	34	37	41	44	47	50	53
12	.05	5	9	13	17	21	26	30	34	38	42	47	51	55	60	64	68	72	77
	.01	2	5	8	11	14	17	21	24	28	31	35	38	42	46	49	53	56	60
13	.05	6	10	15	19	24	28	33	37	42	47	51	56	61	65	70	75	80	84
	.01	2	5	9	12	16	20	23	27	31	35	39	43	47	51	55	59	63	67
14	.05	7	11	16	21	26	31	36	41	46	51	56	61	66	71	77	82	87	92
	.01	2	6	10	13	17	22	26	30	34	38	43	47	51	56	60	65	69	73
15	.05	7	12	18	23	28	33	39	44	50	55	61	66	72	77	83	88	94	100
	.01	3	7	11	15	19	24	28	33	37	42	47	51	56	61	66	70	75	80
16	.05	8	14	19	25	30	36	42	48	54	60	65	71	77	83	89	95	101	107
	.01	3	7	12	16	21	26	31	36	41	46	51	56	61	66	71	76	82	87
17	.05	9	15	20	26	33	39	45	51	57	64	70	77	83	89	96	102	109	115
	.01	4	8	13	18	23	28	33	38	44	49	55	60	66	71	77	82	88	93
18	.05	9	16	22	28	35	41	48	55	61	68	75	82	88	95	102	109	116	123
	.01	4	9	14	19	24	30	36	41	47	53	59	65	70	76	82	88	94	100
19	.05	10	17	23	30	37	44	51	58	65	72	80	87	94	101	109	116	123	130
	.01	4	9	15	20	26	32	38	44	50	56	63	69	75	82	88	94	101	107
20	.05	11	18	25	32	39	47	54	62	69	77	84	92	100	107	115	123	130	138
	.01	5	10	16	22	28	34	40	47	53	60	67	73	80	87	93	100	107	114

FIGURE B.1: Critical values for Mann-Whitney U test (one-tailed testing)

n_2	α	n_1																	
		3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	.05	--	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
	.01	--	0	0	0	0	0	0	0	0	1	1	1	2	2	2	2	3	3
4	.05	--	0	1	2	3	4	4	5	6	7	8	9	10	11	11	12	13	14
	.01	--	--	0	0	0	1	1	2	2	3	3	4	5	5	6	6	7	8
5	.05	0	1	2	3	5	6	7	8	9	11	12	13	14	15	17	18	19	20
	.01	--	--	0	1	1	2	3	4	5	6	7	7	8	9	10	11	12	13
6	.05	1	2	3	5	6	8	10	11	13	14	16	17	19	21	22	24	25	27
	.01	--	0	1	2	3	4	5	6	7	9	10	11	12	13	15	16	17	18
7	.05	1	3	5	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
	.01	--	0	1	3	4	6	7	9	10	12	13	15	16	18	19	21	22	24
8	.05	2	4	6	8	10	13	15	17	19	22	24	26	29	31	34	36	38	41
	.01	--	1	2	4	6	7	9	11	13	15	17	18	20	22	24	26	28	30
9	.05	2	4	7	10	12	15	17	20	23	26	28	31	34	37	39	42	45	48
	.01	0	1	3	5	7	9	11	13	16	18	20	22	24	27	29	31	33	36
10	.05	3	5	8	11	14	17	20	23	26	29	33	36	39	42	45	48	52	55
	.01	0	2	4	6	9	11	13	16	18	21	24	26	29	31	34	37	39	42
11	.05	3	6	9	13	16	19	23	26	30	33	37	40	44	47	51	55	58	62
	.01	0	2	5	7	10	13	16	18	21	24	27	30	33	36	39	42	45	48
12	.05	4	7	11	14	18	22	26	29	33	37	41	45	49	53	57	61	65	69
	.01	1	3	6	9	12	15	18	21	24	27	31	34	37	41	44	47	51	54
13	.05	4	8	12	16	20	24	28	33	37	41	45	50	54	59	63	67	72	76
	.01	1	3	7	10	13	17	20	24	27	31	34	38	42	45	49	53	56	60
14	.05	5	9	13	17	22	26	31	36	40	45	50	55	59	64	67	74	78	83
	.01	1	4	7	11	15	18	22	26	30	34	38	42	46	50	54	58	63	67
15	.05	5	10	14	19	24	29	34	39	44	49	54	59	64	70	75	80	85	90
	.01	2	5	8	12	16	20	24	29	33	37	42	46	51	55	60	64	69	73
16	.05	6	11	15	21	26	31	37	42	47	53	59	64	70	75	81	86	92	98
	.01	2	5	9	13	18	22	27	31	36	41	45	50	55	60	65	70	74	79
17	.05	6	11	17	22	28	34	39	45	51	57	63	67	75	81	87	93	99	105
	.01	2	6	10	15	19	24	29	34	39	44	49	54	60	65	70	75	81	86
18	.05	7	12	18	24	30	36	42	48	55	61	67	74	80	86	93	99	106	112
	.01	2	6	11	16	21	26	31	37	42	47	53	58	64	70	75	81	87	92
19	.05	7	13	19	25	32	38	45	52	58	65	72	78	85	92	99	106	113	119
	.01	3	7	12	17	22	28	33	39	45	51	56	63	69	74	81	87	93	99
20	.05	8	14	20	27	34	41	48	55	62	69	76	83	90	98	105	112	119	127
	.01	3	8	13	18	24	30	36	42	48	54	60	67	73	79	86	92	99	105

FIGURE B.2: Critical values for Mann-Whitney U test (two-tailed testing)

Appendix C

2018 example NLU annotations


```
"nlu": {
    "annotations": {
        "processed_text": "tell me about Starbucks",
        "sentiment": {
            "neg": 0.0,
            "neu": 1.0,
            "pos": 0.0,
            "compound": 0.0
        },
        "ner": {
            "ORGANIZATION": [
                "Starbucks"
            ]
        },
        "bot_ner": {
            "ORGANIZATION": [
                "World Of Warcraft",
                "Super Mario Odyssey"
            ]
        },
```

```
"bot_entities": {
    "world of warcraft": {
        "score": 1,
        "span": {
            "startOffset": 142,
            "endOffset": 159,
            "span": "world of warcraft"
        },
        "entityLink": {
            "identifier": "wd:Q131007",
            "kb": "WIKIDATA",
            "types": [
                "wd:Q7889",
                "wd:Q2249149",
                "wd:Q18593264",
                "wd:Q51938570",
                "wd:Q166142",
                "wd:Q386724",
                "wd:Q11410"
            ],
            "properties": {}
        },
        "entity": "World_Of_Warcraft",
        "age": 1
    }
},
"anaphora_cand": {
    "values": {
        "ORGANIZATION": [
            "World Of Warcraft",
            "Super Mario Odyssey"
        ]
    },
```

```
            "age": {
                "ORGANIZATION": 1
            }
        },
        "intents": {
            "intent": "tell_me_about",
            "param": "Starbucks"
        },
        "postag": [
            [
                "tell",
                "VB"
            ],
            [
                "me",
                "PRP"
            ],
            [
                "about",
                "IN"
            ],
            [
                "Starbucks",
                "NNS"
            ]
        ],
        "nps": [
            "Starbucks"
        ]
    },
    "modules": {
        "processed_text": [
            "Preprocessor",
```

```
            "Truecaser",
            "NERAnaphoraResolution",
            "TalkAboutEntityTransformer",
            "TellMeAboutNormaliser"
        ],
        "sentiment": [
            "VaderNLTK"
        ],
        "ner": [
            "NEREnsemble",
            "NERAnaphoraResolution"
        ],
        "bot_ner": [
            "NEREnsemble"
        ],
        "bot_entities": [
            "EntityLinker"
        ],
        "anaphora_cand": [
            "NERAnaphoraResolution"
        ],
        "intents": [
            "RegexIntents",
            "TalkAboutEntityTransformer",
            "PersonaRegexTopicClassifier"
        ],
        "postag": [
            "MorphoTagger"
        ],
        "nps": [
            "NPDetector"
        ]
    },
```

```
    "processed_text": "tell me about Starbucks"
},
```


LISTING C.1: Mercury-NLU example annotations. In this example, the system's previous response was *" "I love video games too! I think that Super Mario Odyssey is a really fun game! Okay. So *username* what is a video game that you like? I love world of warcraft."* and the user followed up with *"Tell me about Starbucks"*

Bibliography

Angus Addlesee, Arash Eshghi, and Ioannis Konstas. Current challenges in spoken dialogue systems and why they are critical for those living with dementia. *arXiv preprint arXiv:1909.06644*, 2019.

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a human-like open-domain chatbot, 2020a.

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020b.

Alekh Agarwal, Olivier Chapelle, Miroslav Dudík, and John Langford. A reliable effective terascale linear learning system. *The Journal of Machine Learning Research*, 15(1):1111–1133, 2014.

James Allen and Mark Core. Damsl: Dialogue act markup in several layers (draft 2.1). In *Technical Report, Multiparty Discourse Group, Discourse Resource Initiative*, 1997.

J.L. Austin. *How to Do Things with Words*. William James lectures. Clarendon Press, 1975. ISBN 9780198245537. URL https://books.google.co.uk/books?id=7EZz8MRLzVcC.

Rafael E Banchs and Haizhou Li. Iris: a chat-oriented dialogue system based on the vector space model. In *Proceedings of the ACL 2012 System Demonstrations*, pages 37–42, 2012.

Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. *arXiv preprint cs/0405039*, 2004.

Richard Bellman. Dynamic programming and stochastic control processes. *Information and control*, 1(3):228–239, 1958.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

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

Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. 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, 2015. ACM.

Daniel G Bobrow, Ronald M Kaplan, Martin Kay, Donald A Norman, Henry Thompson, and Terry Winograd. Gus, a frame-driven dialog system. *Artificial intelligence*, 8(2):155–173, 1977.

Stefan Bordag. A comparison of co-occurrence and similarity measures as simulations of context. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 52–63. Springer, 2008.

Herve A Bourlard and Nelson Morgan. *Connectionist speech recognition: a hybrid approach*, volume 247. Springer Science & Business Media, 2012.

Susan E Brennan and Michael F Schober. How listeners compensate for disfluencies in spontaneous speech. *Journal of Memory and Language*, 44(2):274–296, 2001.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020a.

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

Joshua W Buck, Saverio Perugini, and Tam V Nguyen. Natural language, mixed-initiative personal assistant agents. In *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*, pages 1–8, 2018.

Justine Cassell and Hannes Vilhjálmsson. Fully embodied conversational avatars: Making communicative behaviors autonomous. *Autonomous agents and multi-agent systems*, 2(1):45–64, 1999.

Alessandra Cervone, Evgeny Stepanov, and Giuseppe Riccardi. Coherence Models for Dialogue. *arXiv:1806.08044 [cs]*, June 2018. URL http://arxiv.org/abs/1806.08044. arXiv: 1806.08044.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35, 2017.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

Herbert H Clark. *Using language*. Cambridge university press, 1996.

Herbert H Clark and Susan E Brennan. Grounding in communication. 1991.

Baiyun Cui, Yingming Li, Yaqing Zhang, and Zhongfei Zhang. Text coherence analysis based on deep neural network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2027–2030, 2017.

Amanda Cercas Curry and Verena Rieser. A crowd-based evaluation of abuse response strategies in conversational agents. *arXiv preprint arXiv:1909.04387*, 2019.

Amanda Cercas Curry, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xinnuo Xu, Ondrej Dušek, Arash Eshghi, Ioannis Konstas, Verena Rieser, et al. Alana v2: Entertaining and informative open-domain social dialogue using ontologies and entity linking. In *1st Proceedings of Alexa Prize (Alexa Prize 2018)*, 2018.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in Imagined Conversations: A New Approach to Understanding Coordination of Linguistic Style in Dialogs. In *Proc. CMCL*, pages 76–87, 2011.

Kees van Deemter, Mariët Theune, and Emiel Krahmer. Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24, 2005. doi: 10.1162/0891201053630291.

Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Christian Dondrup, Ioannis Papaioannou, Jekaterina Novikova, and Oliver Lemon. Introducing a ROS based planning and execution framework for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI International Workshop on Investigating Social Interactions with Artificial Agents*, ISIAA 2017, pages 27–28, 2017. doi: 10.1145/3139491.3139500. URL http://doi.acm.org/10.1145/3139491.3139500.

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

Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar Zaiane. Evaluating Coherence in Dialogue Systems using Entailment. In *Proceedings of the 2019 Conference of the North*, pages 3806–3812, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1381. URL http://aclweb.org/anthology/N19-1381.

Micha Elsner and Eugene Charniak. Coreference-inspired coherence modeling. In *Proceedings of ACL-08: HLT, short papers*, pages 41–44, 2008.

Micha Elsner and Eugene Charniak. Extending the entity grid with entity-specific features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 125–129, 2011.

Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, Hamed Ketabdar, and Sebastian Möller. Modeling user satisfaction with hidden markov models. In *Proceedings of the SIGDIAL 2009 Conference*, pages 170–177, 2009.

Eyal Even-Dar. *Learning rates for Q-learning*. 2001.

Katja Filippova and Michael Strube. Extending the entity-grid coherence model to semantically related entities. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)*, pages 139–142, 2007.

Charles J. Fillmore. Frames and the semantics of understanding. *Quaderni di Semantica*, 6(2):222–254, 1985.

Asbjørn Følstad and Cameron Taylor. Investigating the user experience of customer service chatbot interaction: a framework for qualitative analysis of chatbot dialogues. *Quality and User Experience*, 6(1):1–17, 2021.

Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. Bootstrapping dialog systems with word embeddings. In *Nips, modern machine learning and natural language processing workshop*, volume 2, 2014.

Mary Ellen Foster, Rachid Alami, Olli Gestranius, Oliver Lemon, Marketta Niemelä, Jean-Marc Odobez, and Amit Kumar Pandey. The mummer project:

Engaging human-robot interaction in real-world public spaces. In *International Conference on Social Robotics*, pages 753–763. Springer, 2016.

Mary Ellen Foster, B. Craenen, Amol A. Deshmukh, Oliver Lemon, Emanuele Bastianelli, Christian Dondrup, Ioannis V. Papaioannou, Andrea Vanzo, Jean-Marc Odobez, Olivier Canévet, Yuanzhouhan Cao, Weipeng He, Ángel Martínez-González, Petr Motlícek, Rémy Siegfried, Rachid Alami, Kathleen Belhassein, Guilhem Buisan, Aurélie Clodic, Amandine Mayima, Yoan Sallami, Guillaume Sarthou, Phani Teja Singamaneni, Jules Waldhart, Alexandre Mazel, Maxime Caniot, Marketta Niemelä, Päivi Heikkilä, Hanna Lammi, and Antti Tammela. Mummer: Socially intelligent human-robot interaction in public spaces. *ArXiv*, abs/1909.06749, 2019.

Jamie Fraser, Ioannis Papaioannou, and Oliver Lemon. Spoken conversational ai in video games: Emotional dialogue management increases user engagement. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 179–184, 2018.

Pascale Fung and Grace Ngai. One story, one flow: Hidden markov story models for multilingual multidocument summarization. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(2):1–16, 2006.

Jianfeng Gao, Michel Galley, Lihong Li, et al. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298, 2019.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models, 2020.

Sarik Ghazarian, Ralph Weischedel, Aram Galstyan, and Nanyun Peng. Predictive engagement: An efficient metric for automatic evaluation of open-domain dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7789–7796, 2020.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. A knowledge-grounded neural conversation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

C. J. Gilbert and Erric Hutto. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *AAAI Conference on Weblogs and Social Media*, pages 216–225, Ann Arbor, MI, USA, 2014.

Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

Herbert P Grice. Logic and conversation. In *Speech acts*, pages 41–58. Brill, 1975.

Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986. URL https://aclanthology.org/J86-3001.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2): 203–225, 1995. URL https://aclanthology.org/J95-2003.

Camille Guinaudeau and Michael Strube. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 93–103, 2013.

Prakhar Gupta, Shikib Mehri, Tiancheng Zhao, Amy Pavel, Maxine Eskenazi, and Jeffrey P Bigham. Investigating evaluation of open-domain dialogue systems with human generated multiple references. *arXiv preprint arXiv:1907.10568*, 2019.

Melita Hajdinjak and France Mihelič. The paradise evaluation framework: Issues and findings. *Computational Linguistics*, 32(2):263–272, 2006.

Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. *Cohesion in english*. Routledge, 2014.

Sunao Hara, Norihide Kitaoka, and Kazuya Takeda. Estimation method of user satisfaction using n-gram-based dialog history model for spoken dialog system. In *LREC*, 2010.

Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

Peter A. Heeman, Fan Yang, Andrew L. Kun, and Alexander Shyrokov. Conventions in human-human multi-threaded dialogues: A preliminary study. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, IUI '05, pages 293–295, New York, NY, USA, 2005. ACM. ISBN 1-58113-894-6. doi: 10.1145/1040830.1040903. URL http://doi.acm.org/10.1145/1040830.1040903.

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

Ryuichiro Higashinaka, Yasuhiro Minami, Kohji Dohsaka, and Toyomi Meguro. Issues in predicting user satisfaction transitions in dialogues: Individual differences, evaluation criteria, and prediction models. In *International Workshop on Spoken Dialogue Systems Technology*, pages 48–60. Springer, 2010.

Matthew Hoffman, Francis Bach, and David Blei. Online learning for latent dirichlet allocation. *advances in neural information processing systems*, 23:856–864, 2010.

Graham Hole. Eight things you need to know about interpreting correlations, 2014. URL http://users.sussex.ac.uk/~grahamh/RM1web/Eight%20things%20you%20need%20to%20know%20about%20interpreting%20correlations.pdf.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. Challenges in building intelligent open-domain dialog systems. *arXiv preprint arXiv:1905.05709*, 2019.

Mahaveer Jain, Gil Keren, Jay Mahadeokar, Geoffrey Zweig, Florian Metze, and Yatharth Saraf. Contextual rnn-t for open domain asr. *arXiv preprint arXiv:2006.03411*, 2020.

Kyle Johnson. *What VP Ellipsis Can Do, and What it Can't, But Not Why*, chapter 14, pages 439–479. John Wiley & Sons, Ltd, 2001. ISBN 9780470756416. doi: https://doi.org/10.1002/9780470756416.ch14. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470756416.ch14.

Andrew Kehler and Andrew Kehler. *Coherence, reference, and the theory of grammar*. CSLI publications Stanford, CA, 2002.

Harry Khamis. Measures of association: how to choose? *Journal of Diagnostic Medical Sonography*, 24(3):155–162, 2008.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*, 2021.

Jakub Konrád, Jan Pichl, Petr Marek, Petr Lorenc, Van Duy Ta, Ondřej Kobza, Lenka Hỳlová, and Jan Šedivỳ. Alquist 4.0: Towards social intelligence using generative models and dialogue personalization. *arXiv preprint arXiv:2109.07968*, 2021.

Andrew L. Kun, Alexander Shyrokov, and Peter A. Heeman. Spoken tasks for human-human experiments: Towards in-car speech user interfaces for multi-threaded dialogue. In *Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '10, pages 57–63, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0437-5. doi: 10.1145/1969773.1969784. URL http://doi.acm.org/10.1145/1969773.1969784.

Oliver Lemon and Alexander Gruenstein. Multithreaded context for robust conversational interfaces: Context-sensitive speech recognition and interpretation of corrective fragments. *ACM Trans. Comput.-Hum. Interact.*, 11(3):241–267, September 2004. ISSN 1073-0516. doi: 10.1145/1017494.1017496. URL http://doi.acm.org/10.1145/1017494.1017496.

Oliver Lemon, Alexander Gruenstein, Alexis Battle, and Stanley Peters. Multi-tasking and collaborative activities in dialogue systems. In *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue - Volume 2*, SIGDIAL '02, pages 113–124, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118121.1118137. URL https://doi.org/10.3115/1118121.1118137.

E. Levin, R. Pieraccini, and W. Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11–23, 2000. doi: 10.1109/89.817450.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019. URL https://arxiv.org/abs/1910.13461.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A Persona-Based Neural Conversation Model. *Proc. ACL*, page 10, 2016a. URL http://arxiv.org/abs/1603.06155.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016b. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL https://aclanthology.org/N16-1014.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep Reinforcement Learning for Dialogue Generation. *arXiv:1606.01541 [cs]*, September 2016c. URL http://arxiv.org/abs/1606.01541. arXiv: 1606.01541.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

Pierre Lison. A hybrid approach to dialogue management based on probabilistic rules. *Computer Speech & Language*, 34(1):232–255, 2015. doi: 10.1016/j.csl.2015.01.001.

Pierre Lison and Jörg Tiedemann. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proc. LREC*, Portorož, Slovenia, 2016.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.

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

Annie Louis and Ani Nenkova. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1157–1168, 2012.

Ryan Lowe, Michael Noseworthy, Iulian V Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. Towards an automatic turing test: Learning to evaluate dialogue responses. *arXiv preprint arXiv:1708.07149*, 2017.

Maria das Graças Bruno Marietto, Rafael Varago de Aguiar, Gislene de Oliveira Barbosa, Wagner Tanaka Botelho, Edson Pimentel, Robson dos Santos França, and Vera Lúcia da Silva. Artificial intelligence markup language: a brief tutorial. *arXiv preprint arXiv:1307.3091*, 2013.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.

Michael McTear. Conversational ai: Dialogue systems, conversational agents, and chatbots. *Synthesis Lectures on Human Language Technologies*, 13(3):1–251, 2020.

Shikib Mehri and Maxine Eskenazi. Usr: An unsupervised and reference free evaluation metric for dialog generation. *arXiv preprint arXiv:2005.00456*, 2020.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013b.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013c.

Ruslan Mitkov. *Anaphora resolution: the state of the art.* Citeseer, 1999.

Paul A Murtaugh. In defense of p values. *Ecology*, 95(3):611–617, 2014.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1238. URL https://aclanthology.org/D17-1238.

Mari Ostendorf, Ashvin Kannan, Steve Austin, Owen Kimball, Richard Schwartz, and J Robin Rohlicek. Integration of diverse recognition methodologies through reevaluation of n-best sentence hypotheses. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*, 1991.

Endang Wahyu Pamungkas. Emotionally-aware chatbots: A survey. *arXiv preprint arXiv:1906.09774*, 2019.

Ioannis Papaioannou and Oliver Lemon. Combining chat and task-based multimodal dialogue for more engaging hri: A scalable method using reinforcement learning. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 365–366, 2017.

Ioannis Papaioannou, A Cercas Curry, Jose L Part, Igor Shalyminov, Xinnuo Xu, Yanchao Yu, Ondrej Dušek, Verena Rieser, and Oliver Lemon. Alana: Social dialogue using an ensemble model and a ranker trained on user feedback. *Proc. AWS re: INVENT*, 2017a.

Ioannis Papaioannou, Christian Dondrup, Jekaterina Novikova, and Oliver Lemon. Hybrid chat and task dialogue for more engaging hri using reinforcement learning. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 593–598. IEEE, 2017b.

Ioannis Papaioannou, Christian Dondrup, and Oliver Lemon. Human-robot interaction requires more than slot filling-multi-threaded dialogue for collaborative tasks and social conversation. In *FAIM/ISCA Workshop on Artificial Intelligence for Multimodal Human Robot Interaction*, pages 61–64, 2018.

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

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

Massimo Poesio, Rosemary Stevenson, Barbara Di Eugenio, and Janet Hitzeman. Centering: A parametric theory and its instantiations. *Computational linguistics*, 30(3):309–363, 2004.

Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, Bill Dolan, Yejin Choi, and Jianfeng Gao. Conversing by reading: Contentful neural conversation with on-demand machine reading. *arXiv preprint arXiv:1906.02738*, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

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

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

E Reiter and R Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87, 1997. doi: 10.1017/s1351324997001502.

Verena Rieser and Oliver Lemon. Simulation-based learning of optimal multimodal presentation strategies from wizard-of-oz data. In *AISB 2008 Convention Communication, Interaction and Social Intelligence*, volume 1, page 42, 2008.

Verena Rieser and Oliver Lemon. *Reinforcement learning for adaptive dialogue systems*. Springer, 2011.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, and Jason Weston. Recipes for building an open-domain chatbot, 2020. URL https://arxiv.org/abs/2004.13637.

Vasile Rus and Mihai Lintean. An optimal assessment of natural language student input using word-to-word similarity metrics. In *International Conference on Intelligent Tutoring Systems*, pages 675–676. Springer, 2012.

Takao Sato and Kenchi Hosokawa. Mona lisa effect of eyes and face. *i-Perception*, 3(9):707–707, 2012. doi: 10.1068/if707.

Florian Schmidt. Generalization in generation: A closer look at exposure bias. *rXiv:1910.00292*, 2019.

Alexander Schmitt and Stefan Ultes. Interaction quality: assessing the quality of ongoing spoken dialog interaction by experts—and how it relates to user satisfaction. *Speech Communication*, 74:12–36, 2015.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. What makes a good conversation? how controllable attributes affect human judgments. *arXiv preprint arXiv:1902.08654*, 2019.

Igor Shalyminov, Ondrej Dusek, and Oliver Lemon. Neural response ranking for social conversation: A data-efficient approach. *CoRR*, abs/1811.00967, 2018. URL http://arxiv.org/abs/1811.00967.

Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.

Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2015.

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

Koustuv Sinha, Prasanna Parthasarathi, Jasmine Wang, Ryan Lowe, William L Hamilton, and Joelle Pineau. Learning an unreferenced metric for online dialogue evaluation. *arXiv preprint arXiv:2005.00583*, 2020.

David R. So, Chen Liang, and Quoc V. Le. The evolved transformer, 2019. URL https://arxiv.org/abs/1901.11117.

Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. Two are better than one: An ensemble of retrieval-and generation-based dialog systems. *arXiv preprint arXiv:1610.07149*, 2016.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *proceedings of the 24th ACM international on conference on information and knowledge management*, pages 553–562, 2015.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374, 2000. URL https://www.aclweb.org/anthology/J00-3003.

Jana Straková, Milan Straka, and Jan Hajič. Open-source tools for morphology, lemmatization, pos tagging and named entity recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, 2014.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

Andrew G Sutton, Richard SBarto. *Reinforcement learning*. MIT Press, 1998.

B Tabachnick and LS Fidell. Using multivariate statistics (ed., vol. 6). boston, amerika, 2013.

Ryota Tanaka, Akihide Ozeki, Shugo Kato, and Akinobu Lee. An ensemble dialogue system for facts-based sentence generation. *arXiv preprint arXiv:1902.01529*, 2019.

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

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

Se-Yun Um, Sangshin Oh, Kyungguen Byun, Inseon Jang, ChungHyun Ahn, and Hong-Goo Kang. Emotional speech synthesis with rich and granularized control. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7254–7258. IEEE, 2020.

Andrea Vanzo, Emanuele Bastianelli, and Oliver Lemon. Hierarchical multi-task natural language understanding for cross-domain conversational ai: HERMIT NLU. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, page to appear, Stockholm, Sweden, September 2019. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Anu Venkatesh, Chandra Khatri, Ashwin Ram, Fenfei Guo, Raefer Gabriel, Ashish Nagar, Rohit Prasad, Ming Cheng, Behnam Hedayatnia, Angeliki Metallinou, Rahul Goel, Shaohua Yang, and Anirudh Raju. On evaluating and comparing conversational agents. *CoRR*, abs/1801.03625, 2018a. URL http://arxiv.org/abs/1801.03625.

Anu Venkatesh, Chandra Khatri, Ashwin Ram, Fenfei Guo, Raefer Gabriel, Ashish Nagar, Rohit Prasad, Ming Cheng, Behnam Hedayatnia, Angeliki Metallinou, et al. On evaluating and comparing open domain dialog systems. *arXiv preprint arXiv:1801.03625*, 2018b.

J. Ventre-Dominey, G. Gibert, M. Bosse-Platiere, A. Farnè, P. F. Dominey, and F. Pavani. Embodiment into a robot increases its acceptability. *Scientific Reports*, 9(1):10083, 2019. doi: 10.1038/s41598-019-46528-7. URL https://doi.org/10.1038/s41598-019-46528-7.

Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

Marilyn Walker and Rebecca Passonneau. Date: a dialogue act tagging scheme for evaluation of spoken dialogue systems. Technical report, AT AND T LABS-RESEARCH FLORHAM PARK NJ, 2001.

Marilyn Walker, Rebecca J Passonneau, and Julie E Boland. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 515–522, 2001a.

Marilyn A Walker, Diane J Litman, Candace A Kamm, and Alicia Abella. Paradise: A framework for evaluating spoken dialogue agents. *arXiv preprint cmp-lg/9704004*, 1997.

Marilyn A Walker, John S Aberdeen, Julie E Boland, Elizabeth Owen Bratt, John S Garofolo, Lynette Hirschman, Audrey N Le, Sungbok Lee, Shrikanth S Narayanan, Kishore Papineni, et al. Darpa communicator dialog travel planning systems: the june 2000 data collection. In *INTERSPEECH*, pages 1371–1374. Citeseer, 2001b.

Marilyn A Walker, Alexander I Rudnicky, John S Aberdeen, Elizabeth Owen Bratt, John S Garofolo, Helen Wright Hastie, Audrey N Le, Bryan L Pellom, Alexandros Potamianos, Rebecca J Passonneau, et al. Darpa communicator evaluation: progress from 2000 to 2001. In *INTERSPEECH*, pages 273–276. Citeseer, 2002.

Richard S Wallace. The anatomy of alice. In *Parsing the Turing Test*, pages 181–210. Springer, 2009.

Benjamin Weiss, Ina Wechsung, Christine Kühnel, and Sebastian Möller. Evaluating embodied conversational agents in multimodal interfaces. *Computational Cognitive Science*, 1(1):1–21, 2015.

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

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*, 2015.

Jason D Williams and Steve Young. Using wizard-of-oz simulations to bootstrap reinforcement-learning based dialog management systems. In *Proceedings of the Fourth SIGdial Workshop of Discourse and Dialogue*, pages 135–139, 2003.

Terry Winograd. Shrdlu: A system for dialog. 1972.

Steve Young. Still talking to machines (cognitively speaking). In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

Steve Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5): 1160–1179, 2013a. doi: 10.1109/jproc.2012.2225812.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5): 1160–1179, 2013b.

Zhou Yu, Leah Nicolich-Henkin, Alan W Black, and Alexander Rudnicky. A wizard-of-oz study on a non-task-oriented dialog systems that reacts to user engagement. In *Proceedings of the 17th annual meeting of the Special Interest Group on Discourse and Dialogue*, pages 55–63, 2016a.

Zhou Yu, Ziyu Xu, Alan W Black, and Alexander Rudnicky. Strategy and policy learning for non-task-oriented conversational systems. In *Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue*, pages 404–412, 2016b.

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

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*, 2019b.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, pages 1–17, 2020.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*, 2017.

Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, (Just Accepted):1–62, 2018.

Yimeng Zhuang, Xianliang Wang, Han Zhang, Jinghui Xie, and Xuan Zhu. An ensemble approach to conversation generation. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 51–62. Springer, 2017.

Barret Zoph and Kevin Knight. Multi-source neural translation. *arXiv preprint arXiv:1601.00710*, 2016.