NEURAL SYSTEMS FOR INFORMATIVE CONVERSATIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Ashwin Paranjape

August 2022

This dissertation is online at: https://purl.stanford.edu/rz437xd9519

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Christopher Manning, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Tatsunori Hashimoto

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Dan Jurafsky

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format.

Abstract

Humans, through deep and expressive conversations, have perfected the art of exchanging information about the world around them seamlessly. But even with the latest NLP methods, chatbots struggle in being informative. In this dissertation, I describe my work on building neural systems for informative conversations.

First, I describe Chirpy Cardinal, our Alexa Prize 2020 Socialbot, that was deployed to tens of thousands of users across the US, and served as a test-bed for an initial system for informative conversations. While we used state-of-the-art models that improved over prior work, they fell short of expectations when deployed in the real-world setting. In particular, our system had two components: a retriever to find conversationally relevant passages from a large corpus (like Wikipedia) and a language generator to weave it into the dialogue with conversational-sounding utterances, and these two components were unable to cohesively work together.

Second, inspired by linguistics literature on conversations, I analyze human-human informative conversations and identify various strategies for acknowledgement, presentation, transition and detail-selection. I also present a case study, where I improve acknowledgements by using conditional mutual information to select better chatbot utterances.

Third, I explore the possibility of learning these strategies from data by jointly training a neural retriever and a neural generator such that they work together cohesively. To train them, we need to know which passages are relevant to the conversation, but the abundant conversational data available for training is not annotated for relevant passages! Our method, HINDSIGHT, uses a posterior retriever to find relevant passages during training. The posterior retriever is jointly trained alongside the original retriever and the generator using the evidence lower bound (ELBo). We find that HINDSIGHT has better inductive biases than existing methods - at inference, the retriever finds more relevant passages and the generator is more grounded in the retrieved passages, resulting in better end-to-end performance. Together, these projects provide a strong practical motivation, rich linguistic guidance and an effective training method for our aim of building neural systems to have deep and topically broad conversations.

Acknowledgements

My journey as a PhD student at Stanford has been nothing short of a thrilling adventure. This adventure was made possible by all the inspiring people I met along the way. They all gifted me life skills that I am thankful for and will cherish forever.

First, I would like to thank my advisor, Chris Manning. In many ways, he is my *sarathi*, the *Krishna* to my *Arjun*. Krishna was too good of a warrior to be allowed to fight in the Mahabharat, but he was allowed to be Arjun's charioteer. Krishna was right beside Arjun throughout the war, guiding him through internal and external turmoil. In fact, their (quite informative) conversation is recorded as the *Bhagavad Gita*, a cherished spiritual, ethical and philosophical text. Chris too was the one person who was in the trenches with me; he knew everything that was happening with my research at all points of time. He believed in me when I myself didn't, and I could rest easy knowing that he was in my corner. He noticed every single detail, whether it be in word tokenization or my next project idea, and taught me how to pay attention to the things that matter. Through his calm, steady and clear guidance, he gave me the greatest gift; he took me from being a researcher to being an *independent researcher*.

I would also like to thank my reading and oral committee – Dan Jurafsky, Tatsu Hashimoto, Monica Lam, Herb Clark – who were instrumental in providing me feedback as well as food for thought for the work in this dissertation.

I am grateful to my close collaborators – Siyan Li, Omar Khattab, Amelia Hardy, Siva Reddy, Abi See, Arun Chaganty – who made research a fun and social activity and taught me important life skills along the way. I am thankful to faculty mentors – Chris Potts, Matei Zaharia, Percy Liang, Raghu Subramanian – for nudging me in the right directions at various points along the journey.

I am also thankful to my mentors prior to joining the PhD program – Jure Leskovec, Robert West, Austin Benson, Sunita Sarawagi, Brian Guenter, Pushpak Bhattacharyya and Krishnendu Chatterjee – who inspired me to do research and helped me hone my (nearly non-existent) research skills.

In the formative years of my PhD, I was lucky to have Abi See, Kathleen Kenealy, Haojun Li, Amelia

Hardy, Peng Qi, Kaushik Sadagopan, Minh Phu and Dilara Soylu as my Chirpy cardinal teammates. This project was a grind, and I am grateful to have an energized group of people alongside. In particular, I could not have done this without Abi, who leaned in by co-leading with her clear thoughts, organized work, exemplary code and eloquent writing. I would also like to thank the next year's Alexa Prize team and in particular Ethan Chi, who made it easy for me to transition away from the Alexa Prize.

It takes a village to raise a child and a research group to raise a PhD student. I am thankful to the welcoming, kind and encouraging community that is the Stanford NLP group. The group has given me a place to call my home in the academic community and makes me feel like I belong here. I am grateful to have crossed paths with Dora Demszky, Dan Iter, Alex Tamkin, Kelvin Guu, Ice Pasupat, Gabor Angeli, Urvashi Khandelwal, Yiwei Luo, Kaitlyn Zhou, Sidd Karamcheti, Rishi Bommasani, Lisa Li and many others. I looked forward to our socials – dinners, hikes, board game nights, movie nights, barbecues, retreats – and serendipitous conversations alike. In particular, I cherish insightful whiteboard discussions with Chris's advisees – Danqi Chen, Peng Qi, Yuhao Zhang, Kevin Clark, Dor Arad, Jenny Hong, Antoine Bosselut, John Hewitt, Shikhar Murty, Eric Mitchell, Yuta Koreeda, Anna Goldie, Ben Newman, Ruth-Ann Armstrong, Xinran Zhao, Moussa Doumbouya, Tolulope Orunremi and Demi Guo. Many of my colleagues are now friends.

My research was aided by funding from Bloomberg, Tencent, DARPA, Samsung and Amazon. I would like to thank them and the anonymous people who helped me: anonymous reviewers, crowd workers and Alexa Prize users. I would also like to thank the helpful and encouraging staff from the CS department, the Bechtel International center and medical care providers, who helped me navigate the many curveballs that life throws at international students.

This journey would not have been possible without my supportive friends – Vipul, Nishanth, Aditya, Vatsal, Rohit, Shubham, Vinaya, Vibhuti, Pooja and many others. I would like to thank them for accepting me despite my idiosyncrasies. I have crossed paths with many friends and acquaintances over the years, and I am grateful for those encounters that have shaped me for who I am.

I want to thank my extended family, my numerous aunts, uncles, cousins, grandfathers and grandmothers. I couldn't have done it without their love; they were always there for me when I needed them. I owe nearly all of me, my nature and personality to my mom, Himagauri, and my dad, Pradeep. My mom provided me with an optimistic and empathetic outlook and my dad provided me with a pragmatic and balanced perspective. The older I grow, the more I find their traits emerging in me. I am thankful to my sister, Harshini, for making me the fun and energetic person that I am today, compared to the sulky couch potato version of myself from before she was born. I am lucky to have found love and companionship with my partner Nandita. I feel lucky to have a person in whom I can confide my greatest fears and rejoice over the smallest victories.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

My goal in this dissertation is to build Natural Language Processing (NLP) systems for informative conversations that achieve two goals: (1) have in-depth conversations and (2) talk about a breadth of topics. Both these aspects have been studied individually: deep conversations on narrow topics and shallow conversations on a wide range of topics. In this dissertation, I study them together. I aim to achieve this goal using neural models (a.k.a. artificial neural networks) that I develop in this dissertation. In this chapter, I situate this dissertation and give an overview. In this section, I motivate building neural systems for informative conversations.

What are informative conversations and why study them? For millennia, humans have used conversations as a tool to understand each other, tell stories, communicate information and get things done. Conversations are a very flexible and expressive tool. But these conversations rarely happen in a vacuum, they are *situated* in the world around us. When talking with others, humans often talk about the people, places and events around them. I use the term "informative conversations" to refer to various types of conversations where participants aim to exchange information about the world with each other. In a study on human-human conversations, I find that roughly half of the words uttered by people are in the context of informing others (see Section 4.1). Thus, informative conversations are a prominent type of conversation and worth studying. In Section 1.1, I discuss some types of conversations and situate my work on informative conversations.

Why build computer systems for informative conversations? We (as humans) need to access information all the time. We tend to keep ourselves aware of the latest news events, get help on assembling furniture, find answers to common questions, hear about how our favorite sports team is faring. Most of these informational needs are being met via computers: we navigate complex menus, we iteratively formulate queries to search for information on the web, we find and subscribe to various sources of curated information, etc. But currently, we are not able to talk with computers the way we talk with other humans. We have to forego our years of expertise in human-human communication and need to learn how to interact with bespoke interfaces. In fact, “search-fu” is used in popular culture to indicate mastery in being able to query search engines. By building systems for informative conversations with machines, we can use natural language to fulfill our informational needs in an intuitive and efficient manner. Next, I argue that both – conversational depth and topical breadth – are important for the systems being built today.

Importance of in-depth conversations. “The act of seeking information is not to describe something you know, but rather something you don’t know.” (Taylor, 1968). Often, people are unclear about what they are looking for in the first place. This act of describing the unknown involves a collaboration with the other speaker where the two speakers coordinate with each other over multiple turns, building up shared knowledge about each other’s understanding and adding novel pieces of information to the conversation. I illustrate these ideas via an example in Section 1.2, Figure 1.2. Human-human conversations allow for various complex phenomena that are essential for gathering information; *they are deep*. On the other hand, our current conversations with machines are rudimentary in comparison. For example, we can ask questions to virtual assistants like Alexa, Google Home and Siri, but they are unable to answer nuanced questions nor respond to follow-up questions. Systems that support in-depth conversations will allow humans to express their informational needs intuitively and enable them to collaborate with computers as easily as they would have with other humans.

The need for broad topical coverage. People, when talking with each other, converse on a wide range of topics like entertainment, sports, business, economics, world affairs, etc., and move fluidly between these topics. Being able to talk about a wide range of topics is essential as it allows humans to synthesize information about diverse topics from various sources. While people do not know as much as the entire internet, web search is a ubiquitous way of interacting with web documents. Users today expect computer systems to utilize all the web’s documents and be able to look for information that was generated by humanity as a whole.

In Section 1.2, I situate this dissertation in terms of the historical progression of systems that operate at different conversational depths and topical breadths.

Achieving these goals with neural systems. Many past conversational systems used handcrafted rules. Even today, the industrial systems deployed in practice are based on intent detection, slot extraction and templated responses (Louvan and Magnini, 2020). While they are initially easy to bootstrap with handwritten branches and templates, even numerous handwritten rules are insufficient to capture the broad range of semantic and syntactic variations of natural language. Neural methods enable flexible and responsive utterance generation and trainable knowledge retrieval which are otherwise not possible with existing methods. In this dissertation, I explore various ways in which we can combine the consistency and interpretability of traditional rule-based systems with the flexibility and responsiveness of neural systems.

The rest of the chapter is structured as follows: I first situate my dissertation w.r.t. conversations in general (Section 1.1) and in terms of the conversational depth and topical breadth (Section 1.2); Then, I briefly describe the thesis contributions in Section 1.3 and provide an overview of each of the chapters in Section 1.4; Finally, in Section 1.5, I draw high level conclusions based on the lessons learned and provide some takeaways on research processes that are relevant to anyone building NLP systems.

1.1 Situating Informative Conversations

To be able to study conversations effectively, researchers reduce their scope by categorizing conversations into various types. This categorization is based on many factors including participants' familiarity with each other, social norms, any presupposed goals of the conversation, etc. For example, conversations between experts and novices (Isaacs and Clark, 1987), between strangers (Godfrey et al., 1992), between people collaborating to find information (González-Ibáñez et al., 2013), etc. There are other ways of categorizing the variations in conversations and I describe them in depth in Section 2.5. Based on these types, conversational analysts and linguists can now study each type in isolation while chatbot designers can build systems that advance machine ability for a specific type.

To situate my work, I use the conversational circumplex (see Figure 1.1) introduced by Yeomans et al. (2022). They categorize conversations based on participant goals along two latent axes: relational and informational. Relational conversations aim to build good relationship, e.g. by being honest, apologizing appropriately, reminiscing, having fun, etc. Informative conversations, correspond to high informational intent with the goal of giving and/or receiving information. To do so, participants may ask questions, give directions, make decisions or brainstorm new ideas. In this dissertation, I focus on informative conversations, by studying strategies for human-human informative conversations in Chapter 4 and building an end-to-end system in Chapter 5.

Can we build informative systems in isolation? No. While researchers can categorize conversations, the conversational participants do not actively think about these types. Realistic conversations are often composed of many types. Even if a conversation is supposed to be of a certain type, as a conversation evolves so does its place in the typology. For example, transactional conversations between a customer and a cashier at the cash counter may evolve into talking about their favorite breads and recipes. A social conversation between two acquaintances might evolve into a mutually beneficial business partnership. Thus, any attempt to study or build systems for a particular type is insufficient when dealing with realistic conversations.

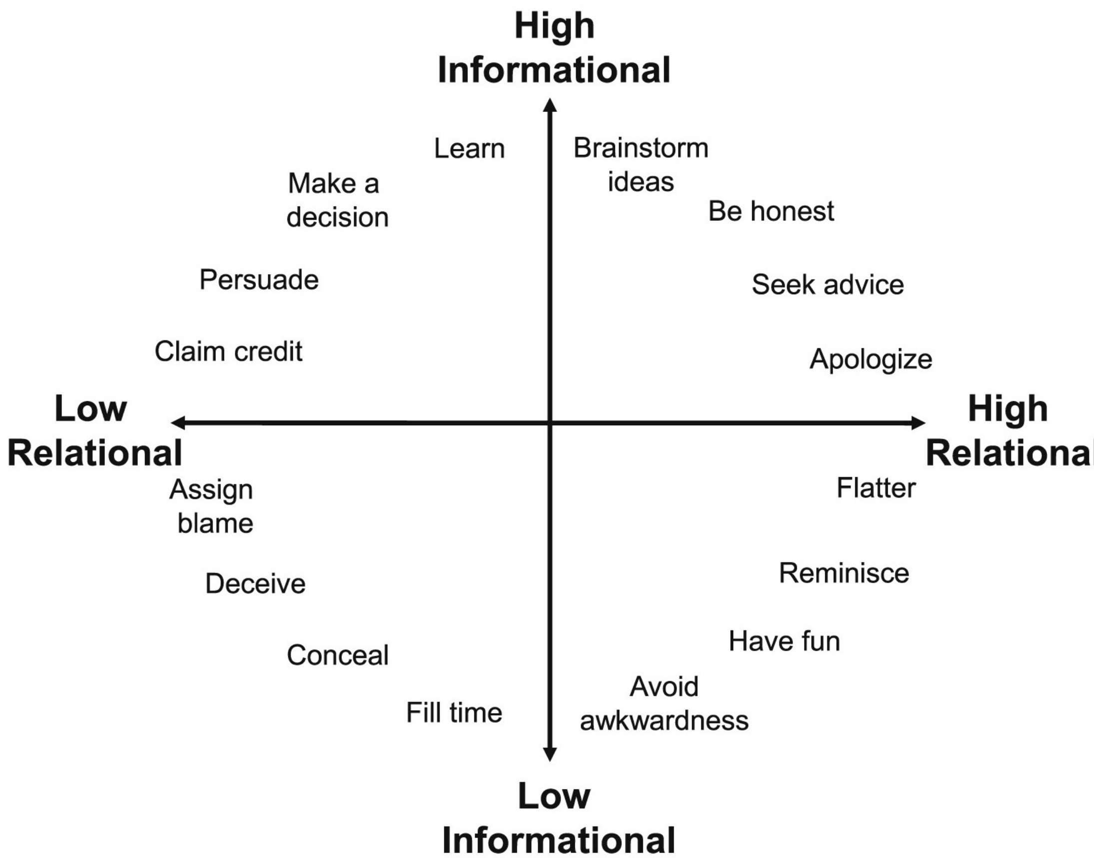


Figure 1.1: **The conversational circumplex by Yeomans et al. (2022) is licensed under CC BY 4.0.** It classifies the many goals people pursue in conversation along two major axes: informational and relational. The focus of this dissertation is on the highly informational types of conversations.

A dialogue system built for a specific type of conversation still has to be able to conduct other types of conversations for it to be a complete experience. Many publicly deployed systems today suffer from their narrow focus. Virtual assistants like Alexa, Google Home or Siri are designed to complete simple tasks: set an alarm, play music, etc. But users often say relevant social utterances that the virtual assistants are not able to handle well, leading to a frustrating experience (Budiu and Laubheimer, 2018). Similarly, many of the text-based chatbot popups that appear as customer service agents on websites railroad the user into the intents they explicitly support. **Therefore, even for systems with a narrow purpose, if they are user-facing, the design is dictated by the human expectation of being able to talk broadly.**

The flip side of frustrating humans by not meeting their expectations is making humans adapt to the system's quirks. When humans are faced with systems that support narrow goals, they avoid *usability agony* by changing their behavior (Whitenton and Budiu, 2018) to the extent that research questions derived from observing such interactions are unlikely to generalize. **Thus, another reason for building complete systems is that they lead to research questions that are ecologically valid.**

While the technical contribution of this dissertation is toward building systems for informative conversations, it is important to recognize and (to some extent) support other types of conversations. Thus, prior to building systems for informative conversations, I build a generic social agent to explore the broader context in which informative conversations arise in Chapter 3 and deploy it to thousands of users across the United States. We wanted the generic social agent (with the design goals as described in Section 3.2) to build good relationships (i.e. be highly relational) and also be informative (i.e. highly informational) while avoiding the kinds of conversations in the bottom left corner of Figure 1.1 (i.e. low relational and low informational). . Furthermore, in Chapter 4, I also analyze human-human informative conversations as they naturally arise in the broader context of two strangers connected over telephone lines. These two projects help ensure that the problems I solve are *ecologically valid* and the designs I propose are *human-centered*.

1.2 Conversational depth and topical breadth

Informative conversations themselves can be of various types. I break them along two major components, topical breadth and conversational depth. Topical breadth is self-explanatory; it simply means being able to talk about many varied topics. In this section I explain conversational depth in detail with examples. Then I justify my position on emulating human-human dialogue for increasing conversational depth, as opposed to creating new interaction paradigms. Finally, I chart historical progress of NLP systems along these two axes and situate my work.

Deep conversations. Having deep conversations is an important aspect of seeking information via dialogue. For example, consider the conversation in Figure 1.2 between Yohan and Iga. Yohan is curious to know about the solar system and Iga describes to him the various objects it is composed of. Out of the various objects, Yohan had not heard of dwarf planets and expresses his desire to know more about them. Then, Iga gives him an example of Pluto which then leads them to talk about how Pluto got demoted to being a dwarf planet from being a planet. At the beginning of the conversation, Yohan did not know what he was looking for and could not have expressed it. But through the conversation, they built up a shared understanding of their beliefs over the course of many turns, allowing him to express his informational needs and explore the unknown. This is an example of a deep conversation and the phenomena observed here are often studied in-depth by linguists

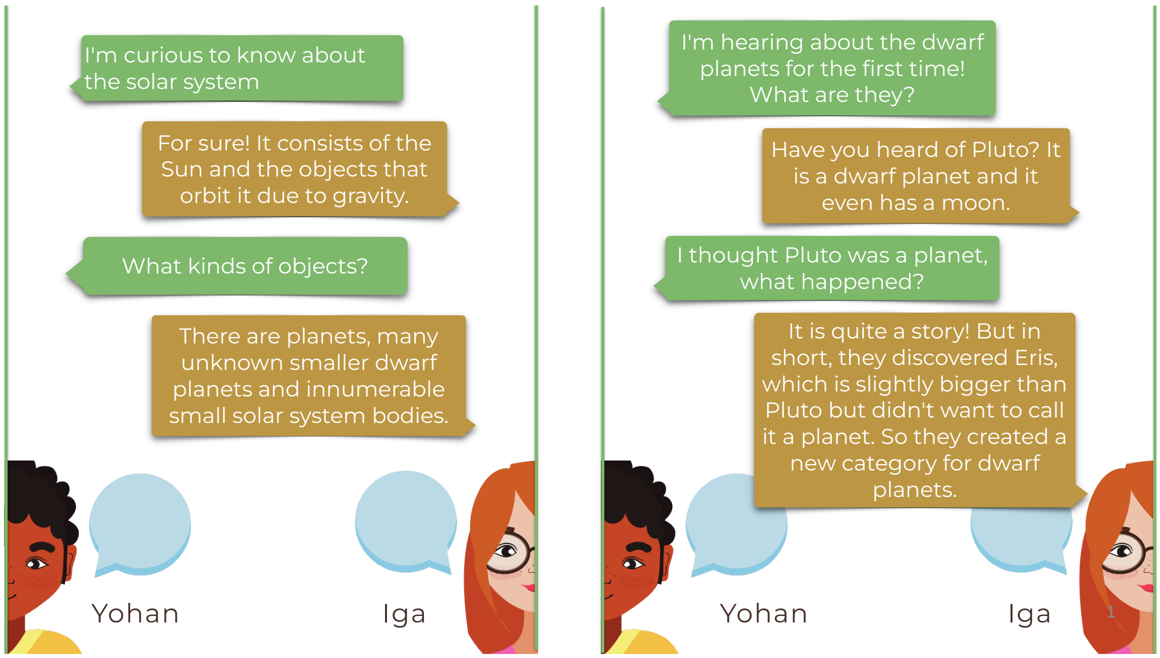


Figure 1.2: **A conversation between Yohan and Iga that illustrates conversational depth.** Initially Yohan does not know what interests him, but as the conversation progresses they collaborate and Yohan is able to explore new knowledge that interests him. This is an example of a deep conversation.

(including psycholinguists and sociolinguists).

Shallow conversations. Contrast this with a hypothetical information-seeking question from Yohan "Which planet was demoted to being a dwarf planet". To be able to ask this question, Yohan needs to know nearly everything except the name of the said planet. I consider this single turn exchange as a shallow conversation. Extracting such answer spans from a corpus of passages is the NLP task of open-domain question answering (Open-QA) (Chen et al., 2017; Kwiatkowski et al., 2019; Petroni et al., 2021) and is a popular and fruitful research direction. However, building systems for shallow conversations is insufficient to capture the phenomena that make human-human conversations flexible and expressive.

Human-human informative conversations are deep, but should human-machine conversations emulate human-human conversations in the first place? There are two points of view. Dahlbäck et al. (1993) argue that humans can adjust their language use when interacting with machines via natural language and therefore the system need not emulate human-human conversations. On the other hand, Reichman (1985); Karis and Dobroth (1991) show that natural language systems (especially spoken dialogue systems) are more usable if they behave in a manner similar to human-human conversations. In my experience, users who repeatedly

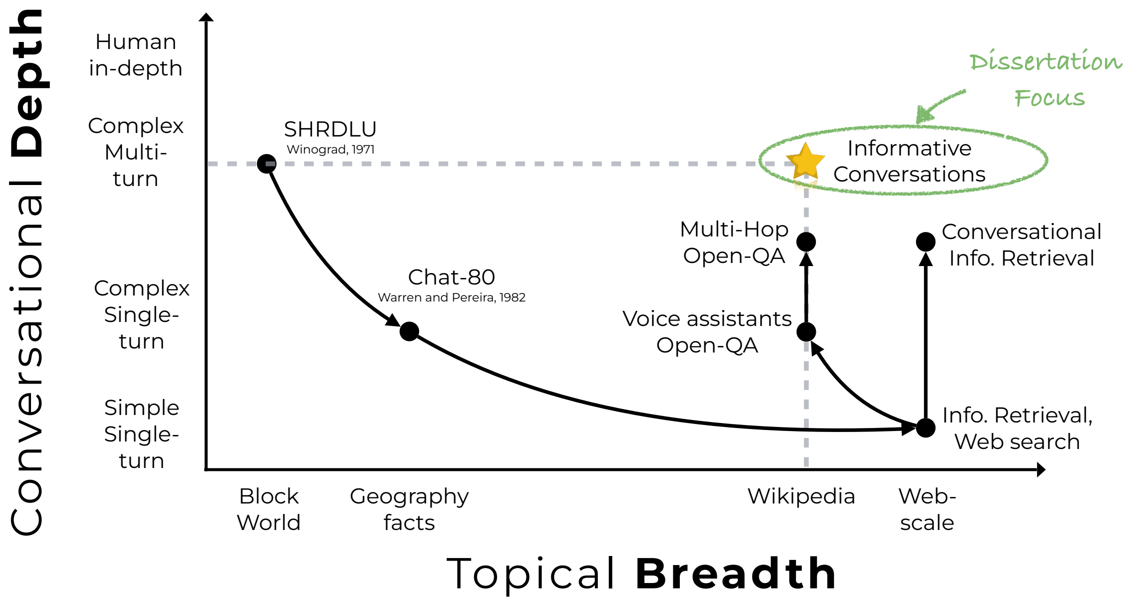


Figure 1.3: **The historical progress of NLP systems.** Many decades ago, we had systems that could have deep conversations on narrow domains. Over time, progress was made in scaling the topical breadth but at the cost of conversational depth. In recent times, the field is making progress in deepening conversations while maintaining topical breadth. In this dissertation I build systems that enable complex and multi-turn conversations at the topical breadth of Wikipedia.

interact with a particular system might be willing to pay the upfront price of understanding common and narrowly-scoped dialogue pathways and adjusting their language to work efficiently with machines. However, skill discovery is a big challenge for spoken interfaces (White, 2018), which do not have the same affordances as visual interfaces. This problem is exacerbated for open-ended dialogue which includes a broad range of variable pathways. I found that users of the Alexa Prize Socialbots were more willing to explore the capabilities of our system and engage with it if they could draw upon their existing ability to have a conversation with another human. For example, a trial system for informative conversations (Section 3.8) allowed the user to explore Wikipedia by providing them section titles as options to select, much like an IVR system. However, users were unwilling to adapt to it, and it was perceived to be a terrible user-experience. This led me to believe in the point-of-view that human-machine conversations should emulate human-human conversations. In this dissertation, I aim to build systems that emulate deep human-human informative conversations.

1.2.1 Situating this dissertation along historical progress

In Figure 1.3, we look at past systems and research areas. Historically, researchers have built systems to have deep conversations, but they were on narrow topics. For instance, SHRDLU (Winograd, 1972) could have a complex multi-turn conversation with a human operator, but was restricted to the block world setting. As an example, the computer would be able to answer "Is at least one of them narrower than the one which I told you to pick up?" by referring to the past conversation and the current state of the world. The system could even interact and update the world. But the block world that it was situated in could only represent blocks of various shapes and colors in various positions. Thus, systems from 50 years ago were having complex multi-turn conversations, albeit on very narrow domains.

Over time, the field of NLP traded off conversational depth for topical breadth. For example, The Chat-80 system (Warren and Pereira, 1982) could answer complex single-turn questions over a few hundred hard-coded geography facts. They translated the questions from English to logical forms, then created a PROLOG program through planning and executed the program to answer the question. While the system operated on a broader and more realistic domain, it was strictly single-turn; it could not remember context from prior questions.

With the web-scale application of information retrieval techniques (Manning et al., 2008), we had systems with unprecedented topical breadth; they could find relevant passages from all the publicly accessible web pages. While these systems were not designed to be conversational, users were nonetheless attempting verbose multi-turn conversations, only to find that the conversational depth was minimal (Radlinski and Craswell, 2017). The interactions were based on matching keywords and restricted to a single-turn (i.e. had no meaningful memory from one search to another). The output of the system was a list of web pages and not a synthesized conversational utterance.

Recent advances in NLP increase conversational depth while retaining web-scale topical breadth. As a prominent example, Open-domain QA and conversational QA are research subfields operating at Wikipedia scale with applications in popular voice assistants such as Alexa, Siri, and Google Home. There has been prior work in information retrieval that argues for "slow search" (Teevan et al., 2013), i.e., trading off speed for higher quality of results. The subfield Conversational Information Retrieval (Gao et al., 2020, 2022), an offshoot of traditional Information Retrieval, aims to make web search more conversational. This dissertation aims to build neural systems that further increase conversational depth to multi-turn complex interactions while operating at the scale of Wikipedia.

1.3 Thesis Contributions

In the previous sections, we situated this dissertation within the broader context of different types of conversations and a historical journey of the trade-off between conversational depth and topical breadth. In this section I describe the key contributions of this dissertation with the underlying journey and the logical progression of research questions in the next section.

The first contribution is a deployment-ready system for social conversations. This system is a hybrid of the traditional rule-based systems and modern neural generation systems. It sets the stage for specialized neural systems for informative conversations.

The second contribution is a set of key strategies used by humans in informative conversations: acknowledgement, transition, detail selection and presentation. These strategies help set the right goals and success indicators for a human facing informative system. I also provide a specific method to extract responses with better acknowledgements (without having to retrain a system) by using conditional mutual information.

The third and central contribution of this thesis is an effective system for extended informative conversations. This system is composed of a special-purpose neural retriever and a special-purpose neural language generator. An important part of this contribution is the training method. Typically, positive passages are needed (Karpukhin et al., 2020; Nguyen et al., 2016) to train a bespoke neural retriever, but in this dissertation I provide a method for training without passage supervision. Passage supervision is also needed to train the neural generator, but in this dissertation I show how a posterior model can be used to mask out passages irrelevant to the desired output.

The end result is a neural system for informative conversations that can be trained from the abundant conversational data available online, adopts many strategies of human informative conversations, fits into a larger architecture for social conversations and is ready for user deployment.

1.4 Thesis Overview

The research conducted as a part of this thesis can be broken down into a three step logical progression: finding important problems, building a perspective with linguistic analysis and finding a solution using ML techniques.

The perfect solution to the wrong problem is not useful. So in the first part of the thesis, I ask the research question, **RQ**$_1$: **"What are the important problems for informative conversations?"**. I aim to find problems based on user needs and limitations of current systems. To do so, I consider a broader set of social conversations and study informative conversations in a natural setting. I co-lead a team of students to build Chirpy Cardinal, our Alexa Prize Socialbot system, that interacted with tens of thousands of users across the

US. This setting is more ecologically valid than evaluating systems via crowd workers because the users were volunteering their time and were intrinsically motivated to chat with a social chatbot. It served as our platform to get real-world conversations and identify the challenges for informative dialogue.

We were one of the first teams to successfully merge the flexibility and responsiveness of neural generation systems with the consistency, control and interpretability of traditional rule-based symbolic dialogue systems. With 9 months of development, our team won 2nd prize, defeating teams that were 3 years in the running. We were also the first team to open source our entire system for use by researchers and industry practitioners. Despite being a successful endeavor, I identified many areas for improvement. In the rest of the thesis, I focus on the subset of challenges for informative dialogue. To provide some context about the system architecture, we used two models for informative dialogue: a retriever to find conversationally relevant content (text passages) and a neural generator (i.e. sequence to sequence model) to write conversational and informative utterances grounded in the retrieved content. Our experience with the Alexa prize competition lead us to two research questions.

First, I realized that the users, based on their lived experience of human-human conversations, had higher expectations from the human-machine conversations which were very visibly subpar. To understand the fine-grained attributes of ideal informative conversations, I ask the next research question **$\mathbf{RQ}_2$ "What strategies do humans employ when talking informatively with other humans?"**. Here, I take inspiration from sociolinguistic and psycholinguistic literature and analyze human-human conversations. I identify 4 sets of strategies: acknowledgement, transitions, detail-selection and presentation. As a case study, I provide a method to improve acknowledgements using conditional mutual information.

Second, I found that the two components of our system, the retriever and the generator, were unaware of each other and therefore did not work well together. The BM25-based retriever found passages based on keyword similarity not based on conversational relevance. This motivated using a trainable neural retriever. On the other hand, the neural generator was a GPT-2 medium model fine-tuned on a parallel corpus of passages and handwritten responses. I found that this kind of supervision was "too clean"; the generator would trust any passage provided to it and include snippets from it without considering relevance and without synthesizing information. Such a generator did not work well in practice because the quality of retrieved passages is variable and relevance to the conversation not guaranteed. I ask the third research question **$\mathbf{RQ}_3$, "How to train a retriever to find conversationally relevant content and a generator to produce grounded utterances such that they work well together?"**. A challenge in training these systems is that we need supervision in the form of passages that help produce output. But the vast majority of conversations available for training do not have this supervision. In the final section of the thesis, I introduce Hindsight, which employs a posterior model to provide passage supervision and jointly train a retriever, which finds conversationally relevant content, and a generator. At inference, the jointly trained retriever and generator are very effective in finding conversationally relevant content and generating grounded utterances. And jointly they outperform existing methods.

In the next subsections, I provide a detailed overview of each part of the thesis.

1.4.1 Problem Finding

To answer **RQ**$_1$: **"What are the important problems for informative conversations?"**, I co-led a team of students to build and deploy a general-purpose social chatbot - Chirpy Cardinal - as an Alexa Prize Socialbot. This work was originally published as *Neural Generation Meets Real People: Towards Emotionally Engaging Mixed-Initiative Conversations* in the Alexa Prize Proceedings (Paranjape et al., 2020).

The Alexa Prize Socialbot Grand Challenge is a competition where university teams can deploy their own chatbots to interact with US-based Alexa users (Gabriel et al., 2020). The conversation begins when users say "Let's chat". In every turn, the Alexa device transcribes the user's utterance to text and sends it to the chatbot. The chatbot produces a text response which is then synthesized into speech by the Alexa device. The conversation continues back and forth for many turns (with good conversations lasting 10 minutes) until the user decides to end the conversation by saying "stop". The user is asked "Would you like to chat with the chatbot again?" and has the option to answer on a scale between 1 and 5.

We designed our chatbot to be user-centric with the following design goals:

1. **Mixed-initiative** : the user should have an equal agency to lead the conversation as the bot and should be able to do so with naturalistic utterances
2. **Empathetic**: the generated utterances should demonstrate empathy when appropriate
3. **Broad-coverage**: the chatbot should be able to discuss niche interests of thousands of diverse users from the US
4. **Conversational**: the utterances produced by the chatbot, particularly when talking about external factual content, should integrate well with the ongoing conversation.

In order to support the above design goals while developing a modular (and maintainable) system, we leveraged the following insight: long conversations are composed of sub-conversations. Each sub-conversation has specific expectations, goals and norms associated with it, and we designed modules called **Response Generators** (RGs) to conduct these sub-conversations. Examples of various RGs are: Personal Chat, Wikipedia, Movies, Music, Opinions, etc. We designed each RG to produce responses for many consecutive turns as

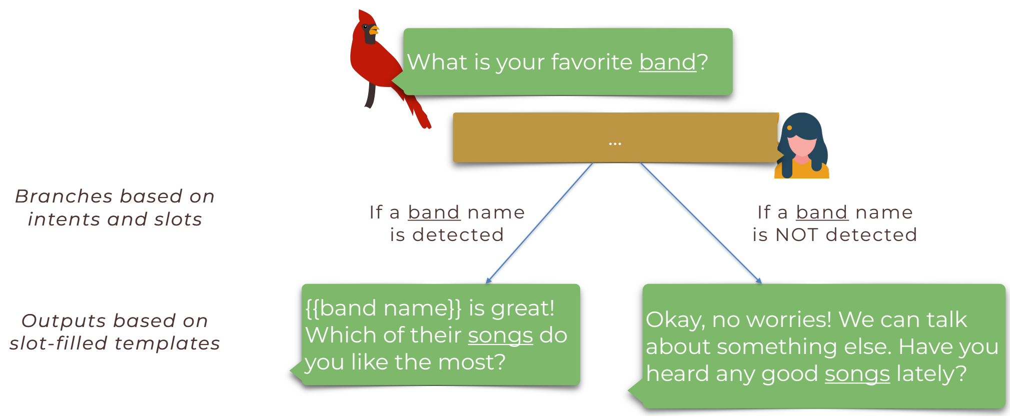


Figure 1.4: **A simplified example of a dialogue tree.** The user utterance is classified based on the detection of a band name and a response is generated by filling slots in templates.

part of the sub-conversation. When the RG self-determines its own inability to continue further, it asks other RGs to take over. Meanwhile, the other RGs are "listening-in" to the conversation and at every turn produce a candidate response if they are able to start a sub-conversation at that turn.

The dominant paradigm for designing modules like RGs is a dialogue tree (or more generally a dialogue graph) as shown in Figure 1.4. The general idea is to first understand the user response (also popularly referred to as Natural Language Understanding or NLU) and classify it into a few branches (Tur and De Mori, 2011). In this phase the system detects intents that identify the user's intention with this utterance and extract slots that are typically objects or named entities needed to query external resources (APIs, knowledge sources). Then the system has to generate a response (also popularly referred to as Natural Language Generation or NLG) and typically they do so by using templates with blanks that are filled using the extracted slots. See Figure 1.4 for an example. These ideas originated from ELIZA which was a rule-based chatbot (Weizenbaum, 1966) and GUS which was frame-based dialogue system (Bobrow et al., 1977).

While this paradigm is easy to bootstrap by handwriting dialogue trees, it does not scale well, especially for open-ended dialogue (see Figure 1.5). The branching logic can be brittle, unable to properly capture syntactic and semantic variations or requiring an ever-growing set of branches. Similarly, the templated utterances are not responsive: they miss out various bits of information not captured in slots. While some issues were resolved using neural networks for intent and slot detection (Qin et al., 2021), we were one of the first teams to deploy artificial neural network based text-generation (a.k.a. neural generation) approaches for more flexible and responsive utterances.

At the core of current neural generation approaches are sequence to sequence (seq2seq) models. They

Figure 1.5: **Examples of failure modes in a dialogue tree.** The failures arise from misclassification (as seen in top row examples) or due to template rigidity (as seen in bottom row examples)

consist of two parts: the encoder, which takes the prior conversation as input along with any auxiliary information and the decoder, which learns a probability distribution over utterances and is responsible for generating the next utterance (Vinyals and Le, 2015; Sordoni et al., 2015; Shang et al., 2015; Li et al., 2016b). During training the model updates its parameters using SGD to increase the likelihood of the next utterance on a given dataset. At inference, tokens are decoded autoregressively, i.e., given a prefix of sampled tokens, the next token is sampled from a distribution induced by the decoder and added to the prefix. There are many popular decoding approaches like greedy decoding, top-K sampling (i.e. sample from top K most probable tokens), beam-search, etc. In this dissertation I typically use nucleus sampling (Holtzman et al., 2020). However, these models when trained from scratch on (relatively) small conversational datasets generate output that is inconsistent even within a single turn.

Most recently, transformer-based seq2seq language models with large parameter counts (>100M) are being "pre-trained" on large scale corpora from the internet (Radford et al., 2018; Raffel et al., 2020). These models serve as a foundation model (Bommasani et al., 2021) for building bespoke models for downstream tasks. In this dissertation, I "finetune" such models (e.g. GPT-2 medium (Radford et al., 2019) and BART-base (Lewis et al., 2020a)) on bespoke conversational datasets (e.g. topical chat (Gopalakrishnan et al., 2019) and Wizard of Wikipedia (Dinan et al., 2019b)). The process of fine-tuning is only slightly different from pre-training: the learning hyperparameters are different, and the dataset is more focussed on the downstream task. These models (of size ~ 350 million parameters) are fast enough to deploy in front of users and can now hold naturalistic and consistent conversations typically up to 3 turns.

However, deploying these neural models out of the box is not sufficient to have longer quality conversation. We identified and attempted to fix 3 problems:

1. **Long-term incoherence**: Since the utterances generated by neural models degrade after 3-5 turns, we stitched multiple such sub-conversations using rules based on symbolic information extracted from prior context. For example, we were able to extract entities and opinions that were mentioned but not the focus of the conversation so far and start a new neural conversation around that entity.

2. **Domain mismatch**: The dialogue datasets (that we use for fine-tuning) are collected under assumptions about the scope and expectation associated with the conversation. For example, the participants in the Empathetic Chats dataset were expected to display empathy in response to various social situations whereas the participants in the Topical chat dataset were expected to introduce fun facts about various topics into the conversation. Neural models trained to emulate these datasets also inherit the scope and tone of the specific dataset, which do not match the overall scope and tone of the larger conversation. To reduce this mismatch we use scripted intros to narrow the scope before launching into the neural sub-conversation, followed by a scripted outro at the end to expand the scope back to the original conversation.

3. **Failures in the interactive setting** When neural systems are deployed to users, the users have complete agency and can say anything, including abruptly changing topics, expressing dissatisfaction, providing feedback, etc. However, the training datasets do not contain such examples and neural models fail under these conditions. We had to detect these cases separately and in the case conversational failure was detected, we would gracefully exit the neurally generated conversation.

Overall our system was successful in engaging users with long and meaningful conversations. Based on thousands of conversations with real people, on average our chatbot was rated 3.6 out of 5 and the top 10% of conversations lasted over 12 minutes. In the finals, our chatbot had 6 conversations with trained conversationalists and came second. Even as a rookie team, with no prior codebase to build upon, we were able to leverage neural methods to iterate quickly and provide a great user experience.

For informative conversations, I developed the Wikipedia Response Generator, which was designed to be able to talk about any topic from Wikipedia. It queried a BM25 retriever using the user's utterance to find relevant passages and a GPT-2 medium model to generate using a retrieved passage. The generator was trained on a parallel corpus of passages and human-written utterances from the Topical Chat dataset. However, I found that our system dialogue was subpar to human dialogue and led to two research questions that I answer in the rest of the dissertation.

1.4.2 Linguistic Perspective

Human conversations (with other humans) have been studied in-depth by sociologists, psychologists and linguists. In this part of the dissertation, I analyze conversations from the Switchboard dataset. Inspired by previously identified phenomena, I find sets of strategies that correspond to successful informative conversations.

My analysis follows the work of Herbert Clark's approach to conversational analysis. I find that people apply four kinds of strategies:

1. **Acknowledgement strategies**: Inspired by Clark and Brennan (1991), I classify all mentions of prior context into various acknowledgement strategies. The major ways to acknowledge prior conversational context are via shared experience, agreement (or disagreement) and back-channeling (common in voice interactions).

2. **Transition strategies**: Inspired by Sacks and Jefferson (1995, Chapter 12, Winter 1971), I identify semantic justifications for topical changes. Nearly half the time, people stay on topic but elaborate their own previous utterance or the other person's previous utterance. More than a quarter of the time they change topics by finding similarities and differences between the two topics.

3. **Presentation strategies**: Inspired by Smith and Clark (1993), I find that people present factual information in non-factual forms like opinions, experiences and recommendations. In fact presenting facts as a part of their experiences and opinions is more common than factual statement.

4. **Detail-selection strategies**: Inspired by Isaacs and Clark (1987), I find that people select the right level of detail depending on the state of the conversation and their belief about the other person's knowledge.

In the second part of Chapter 4, I give a case study on improving acknowledgements using pointwise conditional mutual information. I found that the generation model used in Chirpy Cardinal was sufficiently powerful to generate some samples (out of many) that acknowledge well, so I formulated it as a response selection problem. Here, the conversational history x and relevant new knowledge z are fed as input to the generator model from which $y_1, y_2, \ldots$ are sampled as generations. I provide a method for selecting a sample y_i that contains better acknowledgement.

Prior work by Li et al. (2016a) uses pointwise mutual information (PMI) between samples y_i and prior contexts x, z to find high quality utterances. I find that the sample with Maximum PMI (referred to as the Max-PMI heuristic) typically copies over new knowledge from z and does not acknowledge conversational history x. I propose a new metric, conditional mutual information (pcmi_x) of sample y_i with conversational history x given new information z. This metric better captures the unique overlap of the sample with conversational history. I also propose a heuristic Fused-PCMI that trades off PMI with pcmi_x. I compare between competing methods with human evaluation and find that Fused-PCMI selects responses that acknowledge better and are higher overall quality.

This work was originally published as *Human-like informative conversations via conditional mutual information* in the proceedings of the 2021 Conference of the NAACL-HLT (Paranjape and Manning, 2021).

In the next chapter, I make fundamental changes to the training process to improve other strategies like transition, presentation and detail-selection. I train a neural retriever and a generator to learn human-like strategies latently from data.

1.4.3 ML-based solution

The Chirpy Cardinal system for informative conversations from Chapter 3 was subpar. The retriever failed when the user utterance lacked search-like keywords and even when it did, passages with a maximal term overlap are often different from passages that take the conversation forward. On the other hand, the utterance generator would often ignore the retrieved passages and make up new and incorrect information (i.e. it would hallucinate).

The strategies for human informative conversations from Chapter 4 gave concrete descriptions of ideal retriever and generator behavior. The retriever needs to find passages based on commonalities and differences (to support various transition strategies), passages that support different experiences and opinions and passages that contain varying levels of details. The generator needs to operationalize presentation and detail selection strategies and needs to bridge between topics.

There is abundant conversational data online that is informative and implicitly demonstrates these strategies. Can we train informative systems based on it? We can split a conversation and use the conversation so far as the input x to and train our system to generate the next utterance y as the output. The challenge is that there is no aligned corpus of relevant passages, which are needed to train the retriever and the generator.

Lewis et al. (2020b) attempt to get around this issue by retrieving top-k passages $(z_1, z_2, ..., z_k)$ based on

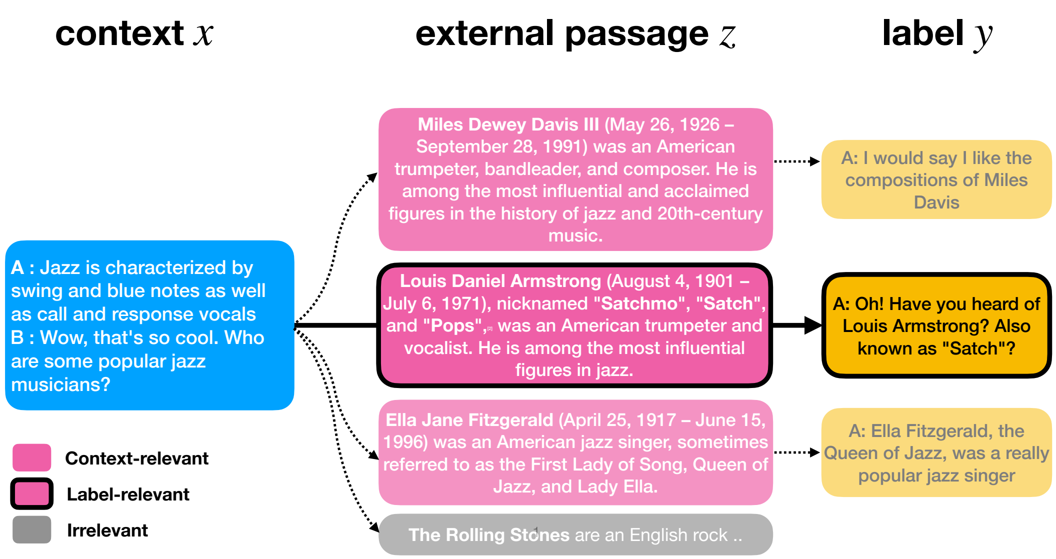


Figure 1.6: **Label-relevant passages are a subset of context-relevant passages for an open-ended conversation.** For a given context x, all context-relevant passages (in pink) are relevant. In this case, all passages in pink are about different jazz musicians, making them context-relevant. However, the next utterance in the training data y only talks about Louis Armstrong and only the passage about Louis Armstrong is useful to product it and is label-relevant. During training, the retriever only has access to x and is therefore unable to separate the label-relevant passage from the context-relevant passages.

the conversational context x and maximizing the marginal approximation:

$$P(y|x) \sim \sum_{z \in \text{top-k}(P_\eta(.|x))} P_\eta(z|x) P_\theta(y|x, z)$$

Here $P_\eta(z|x)$ is the probability distribution over passages z given conversational context x and $P_\theta(y|x, z)$ is the probability distribution over labels y conditioned on conversational context x and a passage z. I found that this method performs poorly for open-ended informative conversations. First, the retriever is bootstrapping itself and is unable to find relevant passages. If the retriever does not find good passages in the top-k retrieved passages, there is little training signal. I find that, even **after** the retriever is trained by this method, it misses out on 45% of relevant passages even when top 100 ($k = 100$) passages are considered (note that typically $k \leq 10$ during optimization). Presumably, the passages found by the retriever were easy to find in the first place and the passages that contained novel training signal were harder to find and therefore missed by the retriever. Secondly, the generator produces utterances that are not grounded in the retrieved passage. For example, in

Figure 1.6 we see that the retriever considers all the passages about jazz musicians to be context-relevant, but when the generator maximizes the likelihood of the observed utterance about Louis Armstrong, it needs to learn to ignore 2 out of the 3 retrieved passages. This happens because the retriever doesn't have enough information to mask out passages that are irrelevant to the label y.

I fix this by using a separate posterior retriever $Q(z|x, y)$ during training that can separate the label-relevant passages from other passages. This retriever has access to y and therefore can find label-relevant passages directly. I find, in our experiments, that it misses out on only 15% of the label-relevant passages compared to 45% from before. Thus, more relevant passages are available during training. The retriever, posterior-guide, and generator are jointly optimized using the evidence lower bound (ELBo):

$$logP(y|x) \geq \mathbb{E}_{z_i \sim Q(.|x,y)}[\log P_\theta(y|x, z)] - D_{\mathrm{KL}}(Q|P_\eta)$$

While the objective function is a lower bound, it encodes biases that improve joint-training on open-ended tasks: (1) conditioning the generator on the passages weighted by their label-relevance (from the label-posterior distribution) increases grounding and (2) training the retriever with a mode-seeking reverse-KL divergence encourages it to match some modes with the guide (label-relevant passages), with a lesser penalty for matching other modes (other context-relevant passages).

Using HINDSIGHT on the Wizard of Wikipedia dataset of informative conversations: the retriever finds more relevant passages with a 23% relative improvement (r.i.) in success@10 (i.e., the label-relevant passage is among the top-10 retrieved passages), the generator is more grounded with 19% r.i. in Novel-F1 overlap with the top-1 retrieved passage (i.e., its overlap with the retrieved passage excluding words that are common or in the input) and the combined system is overall better with a 6.4% r.i. in Novel-F1@1 overlap with the gold utterance (the best matching generation when considering top-1 retrieved passage).

This work was originally published as *Hindsight: Posterior-guided training of retrievers for improved open-ended generation* in ICLR 2022 (Paranjape et al., 2022).

1.5 Learnings

Previous sections situate my work and give a broad overview of this dissertation. In this section, I will highlight high-level learnings from my research. These learnings are not limited to neural models or informative systems, but apply more broadly to the process of conducting human-facing NLP research.

Solving ecologically valid problems. An ecologically valid evaluation setup matches a user's real world context (Hartson and Pyla, 2019). In this dissertation, I observe the behavior of Alexa users when interacting with their devices in a social setting and motivate my research questions. I argue that this is more ecologically valid than observing crowd worker behavior in datasets that are collected via crowdsourcing platforms. Crowd workers are usually given a set of instructions by the researcher which distorts their behavior. Real users are motivated by different objectives than crowd workers and interact differently. Research problems derived from observing crowd worker behavior may not apply to realistic users and I find this to be the case. Only by observing the deployed system, could I recognize its failures and this motivated me to find linguistic strategies that the users were expecting. Prior research did not touch upon these strategies because they were looking at conversations in ecologically invalid situations with distorted user behavior. The lesson here is to start with the user in their natural setting and work backward toward research questions. This takes more work but leads to research solutions for problems faced by real users.

Building a holistic perspective. As a researcher, there is a natural tendency to dive deep into the latest research and ignore past (or even contemporary) discoveries from allied fields. Through conversations with researchers in linguistics and human-computer interaction, I was able to draw upon prior linguistic research to inspire novel research directions in the form of linguistic strategies for informative conversations. Similarly, I was able to draw upon evidence-lower bound (ELBo) as a perfect fit to the problem of jointly training a neural retriever and generator for informative conversations. ELBo as an approximation that is common in the ML community, particularly among people working with reinforcement learning and variational autoencoders. I believe, we stand to benefit by building a more holistic perspective around our research.

Thinking end-to-end. Traditionally, industry favors "pipelined systems" which are composed of multiple models, where the output of the previous one feeds into the next one and the models themselves are unaware of each other. They are favored because they are easier to test, scale and maintain. However, a practitioner working on one component assumes the rest of the system is fixed and that they need to work within those constraints. I argue for thinking about the entire system in an end-to-end fashion.

By "End-to-end neural systems" the research community often refers to a single monolithic model that takes raw input and is trained for the downstream task without any intermediate inputs or outputs. But this is not the interpretation I espouse. I argue that any practical and deployed system needs "windows and knob". A practitioner can peek into the "windows" (for interpretability) to identify issues and turn "knobs" (for controllability) to rectify them.

I argue that as researchers and practitioners we should think of the entire system and the assumptions being made by each component. For instance, a lot of prior research assumed the retriever to be a separate and fixed component. In fact, much of the research on "knowledge selection" (see Section 2.4) simply ranked a set of passages provided by the dataset. However, by realizing that the upstream system is not set in stone (in this case, the retriever), I was able to claim performance gains over methods that used a BM25-based retriever. And even though the Hindsight system is trained end-to-end, it is still composed of two components and provides a "window" and a "knob" in the form of the retrieved passages. This idea of **thinking end-to-end** is generally applicable to any research problem or deployed system.

Chapter 2

Related Work

In this chapter, I will go over some background and work related to my dissertation. For readers from a less computational background, the sections on artificial neural networks (Section 2.1) and large language models (Section 2.2) will introduce the latest computational advances in NLP. For readers who are aware of them, I suggest skipping to Section 2.3 on work related to chatbots and dialogue systems. In Section 2.4, I describe various approaches to knowledge-grounded dialogue in more detail. Readers with a largely computational background will enjoy the insights from the last section (Section 2.5) on connections with fields such as psychology, sociology and linguistics.

2.1 Artificial Neural Networks

Around 2010, models based on neural networks became competitive with the state-of-the-art models in speech and NLP and have since become the de-facto workhorse. Nearly all the models used in this dissertation are based on neural networks. In this section I give a brief overview of the foundations of these models.

Model architectures. While there has been a constant stream of improvements, here I talk about a few milestones that were transformative for the field.

1. **Static Word Embeddings:** Prior to 2013, vector semantics models were used for computing vector representations of words. Afterwards, word2vec (Mikolov et al., 2013a,b), GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2017) provided a computationally efficient way of consuming large quantities of plain (unstructured) text to compute word embeddings. These embeddings were "static" i.e. one vector per word (or subword) in the vocabulary irrespective of its context. Nearly all the future

advances depend on being able to start with well initialized word-vectors, which are used to map the input tokens to vectors before processing them through neural networks. For more details please refer to (Jurafsky and Martin, 2022, Chapter 6).

2. **Recurrent Neural Networks (RNN):** With the availability of more computational power and modern neural network toolkits with automatic differentiation (Al-Rfou et al., 2016), previous ideas about Simple RNNs (Elman, 1990) and LSTMs (Hochreiter and Schmidhuber, 1997) became viable. For a complete explanation, refer (Jurafsky and Martin, 2022, Chapter 9). In an RNN, at each time step a token is consumed from the input and processed alongside a hidden state (or multiple states e.g. cell state and hidden state) to produce an output and a hidden state for the next time step. They can be stacked into layers and can be run bidirectionally as well. RNNs can consume sequences of tokens as an encoder and be used for sequence classification and labelling tasks. They can also autoregressively produce output tokens as a decoder and be used for natural language generation.

3. **Attention-based transformer networks:** With RNNs, the hidden states were a bottleneck, i.e. for information from the first token to reach the neural network processing the last token, it had to pass through a long sequence of hidden states. The next leap in text-processing was inspired by attention networks, first developed for machine translation (Bahdanau et al., 2015). Here each block called a *Transformer* (Vaswani et al., 2017) can directly attend to the hidden states of all tokens via a multi-headed attention mechanism before a multi-layer perceptron (MLP) layer. Since the sequential nature of processing tokens is lost, each token is provided with a position embedding. Additional architectural tricks like residual connections and layer norm help with stability during training. For an extensive explanation, refer (Jurafsky and Martin, 2022, Chapters 9,10)

2.2 Large language models as a foundation

A second revolution happened with the application of these models for pre-training over vast amounts of text data. During pre-training a generic "unsupervised" objective is used to learn a representation of meaning for words or sentences. In the next stage these models are "fine-tuned" for a bespoke downstream task. These models generalize better to downstream tasks, compared to a model trained from scratch for that task, because they have been trained on a large variety of language data. And the same pre-trained model can be used as a foundation model (Bommasani et al., 2021) for many tasks making them broadly useful.

Pre-trained encoder models produce (non-static) contextual embeddings such that the same surface form of a word will have different vector values depending on the context. These models are fine-tuned extensively for sentence classification tasks, and in this thesis we use BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) used as annotators (see in Section 3.5). Pre-trained decoder-only and encoder-decoder transformer models can autoregressively generate text that is of higher quality than recurrent neural networks. In this paper, we fine tune GPT-2 medium (Radford et al., 2019), BART (Lewis et al., 2020a) for various downstream generation tasks as described in sections 3.7, 3.8, 4.2, 5.4.

Pre-training objectives. A key-ingredient of pre-training is the *unsupervised* objective i.e. they are "intrinsic" or "self-contained" to the training data. For pre-training bidirectional encoders, for BERT, Vaswani et al. (2017) use masked-language modelling and next-sentence prediction objectives, and for SpanBERT, Joshi et al. (2020) mask out longer spans of text. On the other hand for pre-training uni-directional decoders, for BART, Lewis et al. (2020a) train an encoder-decoder where the input is noised with token masking, token deletion, sentence permutation, document rotation and span-masking and the expected output is the original text, whereas for T5, Raffel et al. (2020) combine prefix language modelling, BERT-style denoising and deshuffling objectives.

Neural retrieval models. A class of pre-trained models that I use in this thesis are neural retrieval models. They typically produce two dense embeddings: one for the query and one for the document. These are typically produced by bidirectional encoder models like BERT, but pre-trained with different objectives that try to maximize the similarity between a query and the relevant passage. For the ORQA retriever, Lee et al. (2019) pre-train with an Inverse Cloze Task and fine-tune the retriever model with a reader on question-answer pairs. For DPR (dense passage retriever), Karpukhin et al. (2020) pre-train by maximing the negative log-likelihood of the positive passage (a BM25 retrieved passage that contains the answer span) out of a pool of passages containing the positive passage and randomly sampled in-batch negative passages. ColBERT (Khattab and Zaharia, 2020) uses a late-interaction similarity function (MaxSim) and is pre-trained on the MS-MARCO passage ranking task (Nguyen et al., 2016) and ColBERT-QA (Khattab et al., 2021) is further trained using relevance-guided supervision (i.e. the partially trained retriever is used to find hard-negatives as opposed to random in-batch negatives). To exactly find top-k relevant passages, these models need to compute the similarity with all the passages in the retrieval corpus. However, this is computationally prohibitive because it involves computing a dot product of the query vector with millions of document vectors. Instead, in practice the use approximate nearest-neighbour search methods (Jegou et al., 2010; Johnson et al., 2017) that efficiently retrieve an approximately accurate set of top-k passages. In practice, it is common to retrieve many passages (between 10-1000) and rerank them with a model that can potentially use the full cross-attention (Iyer et al., 2021; Lee et al., 2021).

2.3 Chatbots & Dialogue Systems

Dialogue systems have existed since the 1960s. In this section, I will give a brief overview of their historical development and the various methodological approaches taken by subgroups in the community. Much of this classification is after (Jurafsky and Martin, 2022, Chapter 24) and I refer the reader to it for a complete treatment on the subject. Colloquially, *dialogue systems* refer to systems for task-based dialogue and *chatbots* refer to systems for open-ended social or chit-chat conversations. Afterwards, in the second part of this section, I describe some of the recent advances that pertain to this dissertation.

Dialogue systems for task-based dialogue. Task-based dialogue systems aim to help a user solve a task like making a reservation or playing a song. Frame-based dialogue systems (introduced by Bobrow et al. (1977) for travel planning) underlie nearly all modern commercial digital assistants. They typically have a dialogue state based around frames, which are a knowledge structure representing user intents and extracted slots. Further they have if-then rules attached to slots and, separately, production rules to switch control between frames and fill in different frames. More recently, techniques inspired by dataflow synthesis (Andreas et al., 2020) and semantic parsing (Campagna et al., 2019) are being applied to make these systems more robust to real-world use-cases. Newer task-oriented datasets such as MultiWoz (Budzianowski et al., 2018) and Schema-guided dialogue dataset (Rastogi et al., 2020) are being used to train modern neural-network based approaches. Furthermore, because of the structured state-space, these models can be bootstrapped by synthesized data (Campagna et al., 2022) and with self-play, crowdsourcing and online reinforcement learning (Shah et al., 2018).

Chatbots for open-domain conversations. Unlike task-oriented systems, there is no clear objective for open-ended chit-chat or social conversation. For the most part, due to a lack of a clear objective, unlike task-based dialogue systems, there is little planning involved in chatbots. On the other hand, due to a lack of a structured space, it is unclear how to handle a wide-variety of user responses. The primary focus of research on chatbots has historically been on generating a coherent and natural sounding response. ELIZA by Weizenbaum (1966) is an early example of a rule-based chatbot designed to emulate a Rogerian psychologist where one can "assume the pose of knowing nothing of the real world". It had rules that detected the presence of certain words in the user utterance and replied with a templated response. While these rules were not part of a tree (or a graph), there was a memory stack which allowed the chatbot to refer to past utterances. More recently, there has been the rise of corpus-based chatbots that use a corpus of conversations to either (1) retrieve related responses at inference (Jafarpour and Burges, 2010) or (2) train a neural generative model and at inference sample a conditional response autoregressively (Shang et al., 2015; Vinyals and Le, 2015; Sordoni et al., 2015). In Chapter 3, I will introduce Chirpy Cardinal, our Alexa Prize socialbot, with a hybrid architecture (of rule-based and neural/corpus-based systems) which brings some degree of longer-term planning than traditionally thought of by chatbot designers. In 2016, Amazon started conducting yearly Alexa Prize Socialbot Challenges (Ram et al., 2017) with the aim of providing real-world feedback to university teams. We refer the reader to (Huang et al., 2020), who describe the many challenges in building intelligent open-domain dialog systems.

Open-domain Conversational Datasets Often, a special-purpose dialogue corpus is collected by instructing crowd workers to follow certain norms. For instance, the Persona Chat dataset (Zhang et al., 2018) contains conversations between participants who were asked to chat using a provided persona composed of 5 random characteristics. Similarly, the Empathetic Chats dataset (Rashkin et al., 2019) puts one of the participants in a social situation and asks the other participant to have an empathetic conversation. The Commonsense Dialogues dataset (Zhou et al., 2021) is about social contexts where good quality conversations involve commonsense knowledge. On the other hand, the TopicalChat (Gopalakrishnan et al., 2019) and Wizard of Wikipedia (Dinan et al., 2019b) datasets aim to inform the user about topical fun facts and information from Wikipedia respectively, and are two of the datasets used in this work. Most of these datasets typically only looks at short (3-5 turn) conversations. Recently, Xu et al. (2022) collected the Multi-Session Chat (MSC) dataset of 5+ chat sessions between crowdworkers, with each session containing up to 14 utterances.

Chit-chat in task-based dialogue With improvements in open-ended chit-chat (or chatbot) systems, there has been an increased focus on integrating them meaningfully with task-oriented dialogue systems that are widely deployed as virtual assistants. Sun et al. (2021); Chiu et al. (2022) collect datasets representative of a mixture of such conversations and Zhao et al. (2022) also propose a unified dialogue data schema that is compatible with both chit-chat and task-oriented dialogue systems.

Because of the broad mandate of open-domain chatbots (or socialbots), the research community has made progress by focussing on open-ended conversations with narrower mandates. There has been some work on asking clarifying questions (Aliannejadi et al., 2019) and conversational recommendations (Lei et al., 2020). However, a large fraction of the research community has focused on question answering and also conversational question answering.

Conversational Question Answering (QA) Traditionally, this task was referred to as reading comprehension, where given a question and a document containing the answer (Rajpurkar et al., 2016) or not containing the answer (Rajpurkar et al., 2018), the system needed to extract the answer span. Reddy et al. (2019); Choi et al. (2018) expand this setting to be conversational, where one participant has access to a document and the other participant can ask questions about the document via dialogue. This made the setting more realistic and challenging; the systems now needed to attend to prior conversational context and deduce contextual and implied information (the simplest kind being pronoun resolution). In parallel, reading comprehension was then further expanded to Open-domain QA or open-QA for short (Chen et al., 2017; Joshi et al., 2017; Dunn et al., 2017; Kwiatkowski et al., 2019). Here, given a question, the system needed to find the right document and extract the answer span. The conversational setting was similarly expanded by Anantha et al. (2021); Feng (2021) to include the ability to retrieve the passage containing the answer. For details on the various modelling techniques we refer the reader to recent survey papers by Gupta et al. (2020); Zaib et al. (2021).

2.4 Knowledge-grounded dialogue systems

The focus of this dissertation is on *informative conversations*. I consider being informative to be a goal of the conversation and knowledge-grounding as a means to achieve it. There can be ways of being informative without grounding it in knowledge, e.g. by training large language models that memorize knowledge from the training corpus but do not provide any evidence for grounding. On the other hand, even if a response is grounded in a piece of knowledge, it might not be informative. However, the research community often refers to informative conversations as knowledge-grounded dialogue (or background-based dialogue). Here the dialogue systems typically make use of knowledge in some form to produce utterances. There are two main types of knowledge sources: structured (usually as a knowledge graph) and unstructured (i.e. text-based).

Crowdworker-collected Datasets Researchers have collected conversations between two participants where typically one has access to some form of knowledge (referred to as the knowledge pool) that the other does not. The knowledge pool is typically a few (10-60) sentences at a time. The common thread among all these datasets is that they not only contain informative conversations, but also the sentences/passages in which the responses were grounded. For example, in the Wizard of Wikipedia dataset (Dinan et al., 2019b), a crowd worker is allowed to access passages from Wikipedia with a simple (BM25) based retrieval system and acts as the "wizard" while talking to an "apprentice" learning about a topic. The "wizard" also marks the knowledge sentence used (if any). Gopalakrishnan et al. (2019) try to model topical shifts by combining sentences from 3 different but related topics and showing them to the participants. Two other document-grounded datasets are Holl-E (Moghe et al., 2018) and the CMU Document Grounded Dataset (Zhou et al., 2018) which show information about a movie and ask the participants to have a conversation about it.

Showing relevant pieces of knowledge is useful in helping the crowd workers write grounded utterances, but it also makes the conversations unnatural. By looking at the conversations, we can qualitatively see that the crowd workers feel compelled to add information irrespective of its relevance and often with mild paraphrasing of the used knowledge sentences. Furthermore, they are limited to the sentences shown (or retrieved) and therefore the utterances likely carry the biases of how the knowledge pool was created (or retrieved) in the first place.

Naturally-occurring Datasets To remedy this issue Qin et al. (2019) collect a large dataset of Reddit conversations with associated urls and often named-anchors and Ghazvininejad et al. (2018) collect conversations about restaurants from Twitter where the first turn contained a handle or hashtag matching the business on Foursquare. However, these associations do not mean that the utterance is grounded in the linked webpages and even if they are, the supervision they provide is quite coarse (i.e. not at the level of a sentence)

Finally, we refer the reader to the latest datasets used for shared tasks in the workshop on Document-grounded dialogue (DialDoc; Feng et al. (2021, 2022)).

Knowledge Selection Since the datasets were collected by showing a pool of knowledge to crowd workers, then the straightforward NLP task is to select the right piece of knowledge from the given pool, i.e. knowledge selection. Dinan et al. (2019b), in their dataset paper, propose a strong baseline where they encode the dialogue context and passages with a transformer and perform a dot-product attention between the passages. In the hard-attention variant they select the closest matching passage. Liu et al. (2018) use facts about entities similar to the mentioned entities (referred to as "diffusion" in their paper). Li et al. (2019) create a context embedding for the current turn based on the current utterance embedding, current knowledge embedding and prior turn context embeddings (much like RNNs but applied recurrently across turns with a transformer-based encoding in each turn). Meng et al. (2020b) split the task into two: knowledge tracking, i.e. ground the knowledge that has been talked about to the conversation context, and knowledge shifting, i.e. select the knowledge to be talked about next. Recently, Li et al. (2022a) make use of semantic graphs; they automatically convert the background knowledge documents into document semantic graphs and then perform knowledge selection over such graphs. There are many architectural variants (Meng et al., 2020a; Ren et al., 2020; Chen et al., 2020; Kim et al., 2020; Parthasarathi and Pineau, 2018; Zheng et al., 2020; Zhao et al., 2020; Lian et al., 2019; Zhan et al., 2021), and we refer the reader to a Tutorial on Knowledge-Augmented methods for NLP by Zhu et al. (2022) for a more complete treatment.

Knowledge Retrieval As borne out by my experience building a knowledge-grounded informative system in Section 3.8, these methods try to solve a problem that is practically less relevant. In practice, we do not have a knowledge pool to select from, rather we have to retrieve from a large corpus such as Wikipedia. Off-the-shelf retrievers have widely differing behaviors and the knowledge-selection methods are highly dependent on the data distribution that was used to generate the pool in their training set. Furthermore, the off-the-shelf retrievers are very limiting because they find passages based on keyword overlap, but humans incorporate conversationally relevant new knowledge which is often different. Thus, there is a need to build systems that go beyond selection (a.k.a reranking) by including a retriever component suited for the downstream task. Zhang et al. (2022) propose a joint framework for retrieval and grounded text generation using marginalization as described by Lewis et al. (2020b). In this dissertation, I provide a method for knowledge retrieval based on a posterior training scheme.

Grounded-utterance generation Most papers above provide some method for utterance generation. For example, Ghazvininejad et al. (2018) use a Seq2Seq model that conditions on both the conversation history and external facts to generate the response. To enhance grounding, Zheng et al. (2021) use term-level noising of selected knowledge to ensure that the generator learns to focus on certain terms.

Grounding in a knowledge-graph While this is not the focus of this dissertation, there is a lot of interest in the community (especially from industry) to be able to ground in structured knowledge graphs. For example, Yu et al. (2022) use knowledge graphs to augment a fusion-in-decoder architecture for Open-QA and Liu et al. (2021) augmented a pre-trained language generation model (BART) with a knowledge graph. Moon et al. (2019) collect a dataset of conversations where each utterance is manually annotated with reference to entities and paths in a large scale knowledge graph. For task-oriented dialogue, Madotto et al. (2018) incorporate knowledge bases in their end-to-end system using a multi-hop attention mechanism.

Tutoring systems From the fields of education and cognitive sciences, intelligent tutoring systems share many similarities with the systems for informative conversations. Apart from generating responses for input utterances, such systems involve components for learner and domain modelling. They typically cover a curriculum and a set of problems. Through conversation the learners are tasked with applying their knowledge to solve the problem and the conversational script includes hints, misconceptions, prompt questions, etc. Some examples of intelligent tutoring systems are AutoTutor (Graesser et al., 2004), ITSPOKE (Litman et al., 2006), My Science Tutor (Ward et al., 2011). We refer the reader to a survey chapter by D'Mello and Graesser (2013) more details.

2.5 Connections with allied fields

Conversational analysis In the 1960s and early 1970s Harvey Sacks, Emanuel Schegloff, Gail Jefferson, and their students began studying social interaction and established the field of conversational analysis. Since then, sociologists, psychologists and linguists have established a long and robust tradition for categorizing and understanding human conversations under the umbrella term of conversational analysis. Many phenomena from earlier works, such as turn-taking, repair, etc., that are applicable to spoken conversations where audio is produced and consumed are not relevant to this thesis. We refer the reader to an overview of conversational analysis by Sidnell (2016) for more details. In this section I describe related work about various conversational settings, social robots, their anthropomorphism, common ground, people estimating others' knowledge, and sharing and seeking information. Much of the work I describe is influenced by the mechanistic psychology of dialogue proposed by Pickering and Garrod (2004). They propose *the interactive alignment account* whereby the linguistic representations employed by the interlocutors become aligned at many levels simplifying production and comprehension in dialogue. For a deeper exposition of connections with psychology, sociology and linguistics, we refer the reader to books on related topics: Lectures on Conversation by Sacks and Jefferson (1995) and Pragmatics by Levinson (1983).

Variations in conversational setting Many works on conversational analysis choose and analyze a particular setting. For example, there is a body of research analyzing conversations between readers and reference librarians. An example of analyzing such information seeking conversations is by Taylor (1968) who casts them as question-negotiations and finds that users develop their questions through four levels of need: visceral, conscious, formalized and compromised. Francik and Clark (1985) find that when people request for information, they estimate the obstacles (e.g. if the listener might have forgotten) and accordingly make a conditional request (e.g. "Do you remember ..."). Another common setting is task-oriented dialogues and Grosz (1977) describes procedures for identifying shifts in attention focus in this setting.

Social Robots The study of interaction between humans and social robots (of various kinds) provides insights into how conversational agents should be designed. Utilitarian factors like usefulness and adaptability, and hedonic factors like enjoyment, sociability and companionship are important for social robot acceptance (de Graaf and Ben Allouch, 2013). On the other hand, Malinowska (2021) attempts to answer "What Does It Mean to Empathize with a Robot?" primarily from the relational (cooperational) perspective. Perez-Osorio and Wykowska (2020) review literature from philosophy, psychology, human development, culture and human-robot interaction related to predictions concerning other humans' behavior (usually referred to as their mental states, such as beliefs or intentions, a.k.a. intentional stance).

Anthropomorphism of social robots Is it always a good idea to make conversational agents as human-like as possible? I review some work asking this question for social robots. Anthropomorphic agents were associated with greater trust resilience, a higher resistance to breakdowns in trust and incorporating human-like trust repair behavior largely erased differences between gradually deteriorating quality between a human agent, an avatar and a computer (de Visser et al., 2016). Moreover, increased robot social behavior decreased participants' fear and telling users about the robot's role (i.e. framing) generated increased trust in the robot (Groom et al., 2011). However, according to Grimes et al. (2021), people have higher expectations when told an agent is human as opposed to computer. Fink (2012) presents a literature review of anthropomorphism and role in the design of socially interactive robots from social sciences and robotics research, including results from experiments with social robots.

Common Ground Kiesler (2005) describes how the common ground principle of the least collective effort can be used to predict and design human robot interactions. Addressees distinguish shared from private information when interpreting questions during interactive conversation (Brown-Schmidt et al., 2008), and they are effective in marking this distinction in the *form* of their utterances (Heller et al., 2012). *But, is it always better to have shared knowledge between two parties?* According to Wu and Keysar (2007), generally speaking, when two participants have more shared knowledge they use shared terminology to communicate more effectively. However, for new concepts, where there is an opportunity to inform, these participants tend to overestimate each other's common ground and produce terse responses. This causes confusion and makes communication locally ineffective.

Estimating others' knowledge We saw how wrongly estimating others' knowledge causes communication to be locally ineffective. *So how do people estimate others' knowledge?* For conversations between an expert and a layperson, the expert may underestimate or overestimate the layperson's knowledge. If they overestimate, then layperson more often generated questions that reflected comprehension problems; if they underestimate, laypersons asked mainly for additional information previously not addressed in the explanations (Wittwer et al., 2008). To estimate the robot's knowledge, people extrapolate from their own knowledge and from information about the robot's origin and language (Lee et al., 2005). When it comes to humans conversing with artificial partners, humans generally estimate artificial partners to have more knowledge in the task than human partners (Cowan et al., 2017).

Seeking information When it comes to seeking information, people employ many strategies. One such strategy involves using a candidate answer in a query to provide a model of the type of answer that would satisfy the speaker's purpose (Pomerantz, 1988). Separately, studying collaborative information seeking (i.e. two humans are collaboratively using IR systems to search for information) can provide insights for practical informative systems. In this setting, González-Ibáñez et al. (2013) find that remotely located (as opposed to co-located) participants find more diverse information, those using text-based communication were more task-oriented and those with audio support had reduced cognitive load and negative emotions. These insights about the medium of communication can help in designing meaningful and useful conversational interfaces for information seeking.

Chapter 3

Building A System for Social Conversations

3.1 Introduction

In this chapter, we focus on $\mathbf{RQ}_1$: **"What are the important problems for informative conversations?"**. I aim to work with real users in an ecologically valid setting to identify the places where existing systems are limited and unable to meet their expectations. But, people do not have "Informative conversations" for the sake of it. Instead, they have conversations that involve an exchange of information when necessary and appropriate. Thus, I answer $\mathbf{RQ}_1$ by considering social conversations that are broader in scope. I do so by building and deploying a conversational system for it with the help of a team of students that I co-led. In this chapter, I describe our socialbot for open-domain conversation, **Chirpy Cardinal**, built as a research platform during the 2019 Alexa Prize Socialbot competition. Our system won 2nd place in the final evaluation which was based on 18 ratings from expert judges on 6 conversations, along with a $100,000 cash prize. The primary objective of this chapter is to set the stage for problem discovery. A second, but important objective of this chapter, is to describe the nuts and bolts of a practical system so that future researchers and practitioners can learn from the ideas and build upon the components.

What is the Alexa Prize Socialbot Competition? It is a social chatbot competition organized by Amazon Alexa AI where university teams can build and deploy their chatbots to real users (Gabriel et al., 2020). During the competition, US-based Amazon Alexa users could give an invocation phrase (such as *let's chat*) to be connected to one of the competing socialbots (chosen randomly). After receiving a minimal orientation phrase at the beginning of the conversation, the user talks to the socialbot (in English) until they decide to end the conversation – at which point, they are invited to rate (from 1 to 5) and comment.

Design Goals. Our goal in building this socialbot was to offer a natural-sounding and emotionally engaging dialogue agent that can talk knowledgeably about a wide variety of topics, while also letting the user take as much initiative as possible. Concretely, we had 4 user-experience (UX) goals (see Section 3.2) that were central to our design decisions: mixed-initiative interactions, empathy toward the user, conversational styling of informative utterances and a broad topical coverage. The last two goals correspond to the two cornerstones of this dissertation: rich conversations and high topical breadth.

System Architecture. Our key insight (also observed by Clark (2006)) was that long conversations are composed of sub-conversations; each with their own expectations, goals and norms. We designed our architecture around an array of response generators (RGs) that largely aligned with these sub-conversations, as described in Section 3.6. These RGs exist in a larger framework (Section 3.3) supported by a pipeline of standard NLP annotators (Section 3.5) and a dialogue manager (Section 3.4).

Hybridizing neural and rule-based architectures. The advent of large-scale pretrained **neural generative models** has substantially impacted what is possible in open-domain socialbots. While in the previous Alexa Prize competition (i.e. 2019), none of the top three socialbots used neural generation (Chen et al., 2018; Pichi et al., 2018; Curry et al., 2018), we found current GPT-2 models (Radford et al., 2019) to be a key tool to support our design goals. Neural generation enables natural phrasing and emotional engagement, as well as more flexible responsiveness (e.g., when used as a fallback in Section 3.6.5), supporting higher user initiative. A limitation of neural generation methods for dialogue is deterioration in quality and consistency over a long conversation, which can be potentially overcome with symbolic constraints. We explore ways to bring the best of both worlds – long term consistency and short term fluidity – together. We have two special neural RGs: one for empathetic conversation (Neural Chat in Section 3.7) and the other for informative conversations (Wiki in Section 3.8). The Wiki RG and my experience with it sets the stage for rest of my research.

Result summary. Despite being a first-time entrant, at the end of the competition our system achieved an average rating of 3.6/5.0, which is within 0.1 of the highest-ranked systems, and is capable of detailed, sustained conversations with interested users (with a 90th percentile conversation duration of 12 minutes 55 seconds). Qualitatively, during in-person interactions with users, we observed that many innovations such as in-depth discussions of everyday life, conversational styling of informational content, and opinionated exchanges were received with expressions of pleasant surprise – indicating our steps were in the right direction. After the competition, we open-sourced our code and it can be found at the link https://github.com/stanfordnlp/chirpycardinal. Many teams from next year used ideas and components from our technical article and open-sourced codebase. I discuss implications of Chirpy Cardinal for my dissertation in Section 3.11.

3.2 User-experience Goals

To provide a convincing user experience, an open-domain conversational agent must excel at language understanding, language generation, emotional engagement, memory, world knowledge and conversational planning, among other desirable characteristics – an ambitious goal! Prior work within and outside the Alexa Prize competition has taken the successful strategy of pushing progress along individual skills, and forming an ensemble of sub-systems, each excelling at a singular characteristic while ignoring others. For instance, supporting user initiative in open-domain conversations is extremely challenging, as it requires understanding the countless ways a user can take initiative, and the ability to respond to each of them with specificity. Faced with this difficulty, when it comes to in-depth conversations, many previous dialogue systems rely primarily on bot-initiative, driving users along carefully scripted paths. On the other hand, systems attempting higher user-initiative via non-scripted paths are likely to become nonsensical and non sequitur (Hutchens and Alder, 1998). Thus, there is a lot of room for innovation and research in trying to simultaneously achieve two or more complementary characteristics; this is a recurring theme throughout this work.

Initiative – the ability to drive the direction of the conversation – has been studied extensively in the context of task-oriented dialogue. **Mixed initiative** (Horvitz, 1999), in which the user and the bot share initiative, is an important quality of a successful dialogue system, as it provides the user a sense of agency without making them entirely responsible for suggesting new topics and directions. In order to improve on mixed initiative while still providing an acceptable conversational depth, we designed our initial system to rely heavily on system initiative, but at the same time explored several avenues to increase user initiative in a controlled fashion. To support mixed initiative, our system has a global navigational intent classifier (Section 3.4.1) and entity tracker (Section 3.4.2), allowing it to track high level topic changes from both the user and the bot. Further, our response priority system (Section 3.4.3) allows individual Response Generators (RGs) to interject when the user initiates a change of topic.

High-coverage world knowledge is an important component of open-domain conversation – our bot must be able to talk about the diverse range of entities and topics that interest users, particularly if we wish to respect user initiative. We use the Alexa Knowledge Graph, The Washington Post, Reddit and Twitter as sources of up-to-date knowledge in particular domains, while ensuring high coverage by using Wikipedia and Wikidata entities as the foundation of our entity-based conversations (Sections 3.5.4, 3.4.2, 3.9.3). However, world knowledge must be delivered in a **conversational style** – this is a characteristic that distinguishes a socialbot from a virtual assistant. To achieve this, we fine-tuned a neural generative model on the TopicalChat dataset (Gopalakrishnan et al., 2019) to obtain a conversational paraphrasing model that adapts external text into a conversational style (Section 3.8).

A socialbot cannot focus solely on external entities – to be truly *social*, it must be able to discuss **personal experiences and emotions**. While ELIZA-like systems (Weizenbaum, 1966) attempt this via templated repetition of user phrases, they lack the naturalness and depth of real human conversations. Our Neural Chat module (Section 3.7) invites the user to share their everyday experiences and current emotions, and uses a neural generative model to respond empathetically. With it, we attempt to have a deep, sustained and emotionally engaging conversation about users' lives. In addition, our Opinion module (Section 3.6.2) allows the user to express their feelings by expressing their likes and dislikes. To foster a reciprocal atmosphere, our bot also shares its own distinct feelings, experiences and opinions.

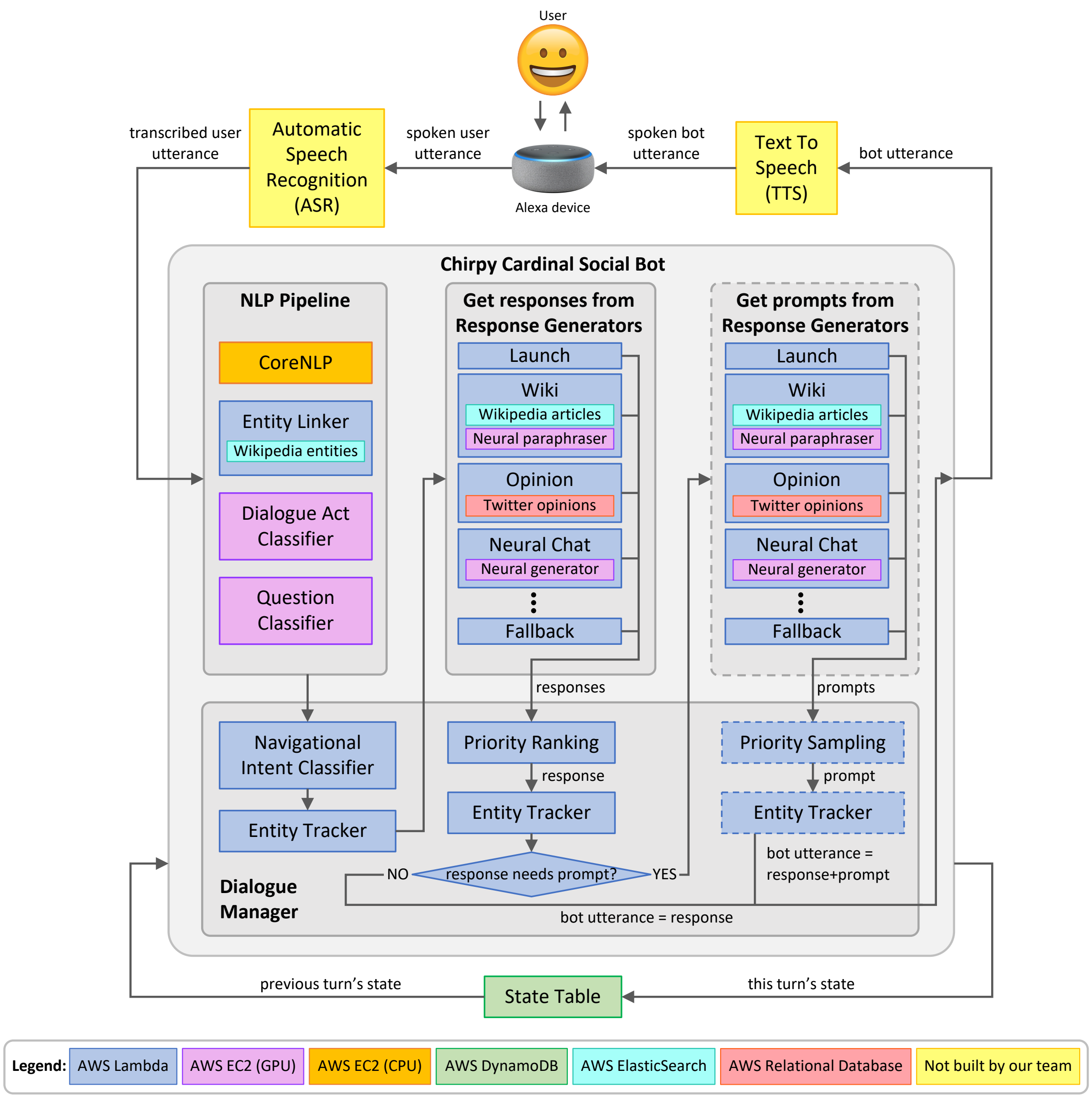


Figure 3.1: **Chirpy Cardinal overall system design.**

3.3 System Overview

Our overall system design is shown in Figure 3.1. At a high level, it is composed of 3 key components: **NLP pipeline**, **response generators (RGs)** and a **dialogue manager**. The NLP Pipeline annotates the incoming user utterance. Based on the user utterance and NLP pipeline's annotations, each RG produces a candidate **response**. If a RG intends to transfer control to a different RG, the old RG asks for a **prompt** from the new RG. The prompt is appended to the selected response for a smoother transition to the new RG. The dialogue manager oversees this entire process, in particular, it determines navigational intents, tracks entities and selects from the candidate responses and prompts. The next paragraphs describe the entire process in more detail.

Our system is built on top of the CoBot framework (Khatri et al., 2018). On each turn, the user's spoken utterance is transcribed by Alexa's Automatic Speech Recognition (ASR) service. The transcribed utterance (which is lowercase, no punctuation) is sent to our AWS Lambda function, which handles the core logic of our bot. AWS Lambda is a serverless computing platform, which means that our function is stateless. To preserve information between turns, we store our bot's overall state in an external State Table (see Figure 3.1), hosted on AWS DynamoDB. At the start of the turn, the previous turn's state is fetched from the table.

We then run the **NLP Pipeline** (see Section 3.5) – a collection of modules that produce annotations based on the user's utterance and the current state. Modules requiring greater computational resources are hosted on remote EC2 instances, while less-demanding modules are hosted within the Lambda function. The NLP Pipeline is organized as a directed acyclic graph (DAG), allowing modules to use other modules' annotations as inputs. To minimize latency, modules are run in parallel where possible, with each module starting as soon as its inputs are ready.

Next, we analyze the user's utterance to determine whether the user wants to talk about any particular entity (see **Navigational Intent**, Section 3.4.1), and update the current entity under discussion if appropriate (see **Entity Tracker**, Section 3.4.2). Finally,

We then run our collection of **Response Generators** (RGs), modules designed to handle particular conversational duties, in parallel (see Section 3.6). Each RG either produces a **response**, or no response (None). If an RG produces a response, it also supplies a **response priority** (see Section 3.4.3), indicates whether the response needs a **prompt** added from another response generator (see Section 3.4.3), and specifies what the current entity under discussion should be, if the response is chosen. The **Priority Ranking** module chooses the response with the highest priority, and the Entity Tracker updates the current entity under discussion accordingly. If the chosen response *does not* need a prompt, it forms the entire bot utterance.

If the chosen response *does* need a prompt, we run our collection of RGs a second time. Each RG either produces a prompt or no prompt (None). If an RG produces a prompt, it also supplies a **prompt priority** (see

Section 3.4.3) and a current entity, as before. The **Priority Sampling** module chooses the prompt by sampling from the supplied prompts, with the probability distribution depending on both the priorities of the prompts and the RGs that produced them. The Entity Tracker updates the current entity again, and the bot's utterance is then formed by appending the prompt to the response.

At the end of the turn, the bot's overall state contains the user's utterance, the conversational history, the NLP Pipeline annotations for the user's utterance, and a state for each individual Response Generator.[1] We write the new state to the State Table, and send the bot utterance to Alexa's Text To Speech (TTS) service, which delivers the spoken bot utterance to the user.

#	Utterance	Entity	Commentary
1	**User:** let's chat	None	
	Bot: Hi, this is an Alexa Prize Socialbot. I'd love to get to know you a bit better before we chat! Is it all right if I ask for your name? [Launch]	None	Launch gives *FORCE_START* response to begin the conversation.
2	**User:** my name is chris	None	
	Bot: Well it's nice to meet you, Chris! I'm excited to chat with you today. [Launch] I hope your afternoon is going well. What are your plans for the rest of today? [Neural Chat]	None	Launch gives a *STRONG_CONTINUE* response that needs a prompt, which is supplied by Neural Chat with *FORCE_START* priority (scripted transition).
3	**User:** hang out with my friends	None	
	Bot: That sounds great, I love spending time with friends. What will you do together? [Neural Chat]	None	Neural Chat generates a GPT-2 response with *STRONG_CONTINUE* priority.
4	**User:** maybe watch a movie	None	
	Bot: Me too! I love watching movies; I get to learn so much about what the world is like outside of the cloud! Have you seen any movies recently? [Movies]	Film	Movies detects the 'movie' keyword in the user's utterance and gives a *FORCE_START* response (overriding Neural Chat), setting current entity to *Film*.
5	**User:** i saw the matrix	The Matrix	The Entity Linker and Entity Tracker identify *The Matrix* as the current entity.
	Bot: Nice! Did you like The Matrix? [Movies]	The Matrix	
6	**User:** i loved it neo is amazing	The Matrix	
	Bot: Oooh, yeah, I agree. Hey, isn't Keanu Reeves in that movie? What do you think about Keanu Reeves? [Movies]	Keanu Reeves	Movies sets current entity to *Keanu Reeves*.
7	**User:** i want to talk about something else	None	Navigational Intent detects *negative intent*, so Entity Tracker rejects the current entity.
	Bot: OK, no problem. [Movies] There's actually something else I wanted to ask you about. What's your favorite animal? [Categories]	Animal	Movies acknowledges and asks for a prompt. Categories gives a *GENERIC* prompt.
8	**User:** i love cats	Cat	Entity Linker and Entity Tracker identify *Cat* as current entity.

[1]Each individual RG state contains information required by that RG – for example, it might contain the current node (aka treelet, see Section 3.6.1) in the RG's dialogue graph, or a list of the utterances and/or entities that have been discussed, to avoid repetition. Each RG maintains its own individual state.

	Bot: Good to hear you like cats. I have to be honest though, I'm not a big fan of cats. I feel like cats don't respect my personal space, but I would love to hear why you like cats? [Opinion]	Cat	Opinion gives a *CAN_START* response about cats.
9	**User:** hmm i love cats because they are fluffy	Cat	
	Bot: That make sense. Now that I think about it, one good reason to like cats is that they purr and I definitely need that kind of positive feedback. [Opinion] Wanna know something interesting about cat? [Wiki]	Cat	Opinion finishes talking. Wiki gives a *CURRENT_TOPIC* prompt about *Cat*.
10	**User:** you are not very smart	Cat	
	Bot: I know you feel frustrated. I'm always trying to get better. [Offensive User] I've been listening to some new music today and I wanted to chat about instruments. If you were a musical instrument which one would you be? [Music]	Musical instrument	Offensive User gives a *FORCE_START* response to the critical utterance, then [Music] provides a *GENERIC* prompt to steer the conversation in a new direction.
11	**User:** what do you find interesting	None	The user gives an unexpected response.
	Bot: I like the violin, but I'm more of a classical music player. [Neural Fallback] I remember you mentioned Neo. Would you like to talk more about it? [Wiki]	Neo (The Matrix)	Neural Fallback gives a GPT-2-generated response, which is selected as no other RG (except Fallback) has a response. Wiki gives a *CONTEXTUAL* prompt.
12	**User:** sure	Neo (The Matrix)	
	Bot: Neo has got into so many adventures! What are some adventures that you've enjoyed? [Wiki]	Neo (The Matrix)	Wiki asks an open-ended question (specific to movie characters) about *Neo (The Matrix)*.
13	**User:** morpheus teaching jujitsu to neo	Neo (The Matrix)	
	Bot: I liked that Neo and Trinity were able to rescue Morpheus from a building protected by armed guards and agents. Morpheus has done so many amazing things! What are some of your favorite Morpheus moments? [Wiki]	Morpheus (The Matrix)	Wiki gives a response (generated by the GPT-2 conversational paraphrasing model), sets the current entity to *Morpheus (The Matrix)*, and asks another open-ended question.
14	**User:** i want to stop talking	None	We detect *stop intent*; the conversation ends.

Table 3.1: **An example dialogue between a user and Chirpy Cardinal.** To respect users' privacy, this is not a real user conversation.

3.4 Dialogue Management

Our Dialogue Manager handles the high-level logic of tracking which topics we are discussing with the user, and which responses (and prompts) should be used to form the bot's utterances. It consists of 3 main components: the navigational intent classifier, the entity tracker, the response-and-prompt system. The response-and-prompt system in turn has two subcomponents: the response priority ranking system and the prompt priority sampling system.

3.4.1 Navigational Intent Classifier

A user has *navigational intent* when they are indicating that they do (*positive*) or do not (*negative*) want to talk about a particular topic. Users might give navigational intent while specifying the topic (*can we talk* *about minecraft*, *stop talking about minecraft*), or referring to the current topic (*let's discuss this more*, *could you change the subject*), or referring to no topic (*alexa can we talk*, *i don't want to chat any more*). Users sometimes give positive and negative navigational intent in the same utterance (*i don't want to talk about movies any more let's chat about you*). To recognize navigational intent, we use manually-constructed regexes, as they are quite high precision.

3.4.2 Entity Tracker

For our response generators to work together to discuss different topics smoothly, we must track which entities we are currently discussing, which we have finished discussing, and possible entities to discuss in the future. This is the role of the *entity tracker*. We assume that at any point in the conversation, there is one *current entity*, which is either a Wikipedia entity (see Section 3.5.4) or None (if we're discussing something that does not have a Wikipedia article (see for instance, Table 3.1 Turn 3)). The current entity is updated at most three times per turn (see Figure 3.1):

1. **After analyzing the user's utterance**. The entity tracker uses the entity linker's output, which is a priority-ordered list of possible entities mentioned by the user on this turn, along with their scores (see Section 3.5.4 for details). If the user expressed negative navigational intent towards the current entity, it is rejected. If the user expressed positive navigational intent towards some topic, we search inside the topic slot in their utterance; the highest-priority entity with score over a low threshold (1,000) is chosen as current entity. If there is a particular type of entity we expect the user to mention on this turn (e.g. if the bot asked *What's your favorite movie?*) and there is an entity with the expected Wikidata category (e.g. *film*) with score over a low threshold (1,000), it is chosen as current entity. Otherwise, if the entity linker has made a prediction with sufficiently high score (over 10,000), it becomes the current entity. If none of these conditions are met, the current entity stays the same.

2. **After choosing the response**. When the RGs provide responses, each RG also specifies what the new current entity should be, if its response is selected by the priority ranker. We update the current entity to be whatever was provided by the selected RG.

3. **After choosing the prompt**. If we get a prompt, we update the current entity similarly.

This system allows the user to initiate topics (e.g. the bot starts talking about cats if the user utterance is *i want to talk about cats*), allows RGs to initiate topics (see Table 3.1, Turn 4), allows multiple RGs to talk seamlessly about the same topic (see Table 3.1, Turn 10), and allows RGs to signal when a topic should be finished (see Table 3.1, Turn 7).

Response Priority	Meaning
FORCE_START	This inactive RG should take control (e.g., Table 3.1, Turn 4), or override, such as handling offensive user utterances (e.g., Table 3.1, Turn 10).
STRONG_CONTINUE	This active RG can continue the conversation with a good next response (e.g., Table 3.1, Turn 2). Only a *FORCE_START* can override it.
CAN_START	This inactive RG can potentially take control (e.g., Table 3.1, Turn 8), but should not interrupt a *STRONG_CONTINUE*.
WEAK_CONTINUE	This active RG can continue the conversation but its next response is of poorer quality. It should be overridden by any available *CAN_START*s (or higher).
UNIVERSAL_FALLBACK	Only used by Fallback and Neural Fallback RGs (e.g., Section 3.6 and Table 3.1, Turn 11)

Table 3.2: **Response Priorities.** (ordered by descending importance)

3.4.3 Response-and-Prompt System

As described in Section 3.3, on some turns the bot utterance consists of a **response** from one RG, followed by a **prompt** from another RG. This system is useful when the responding RG can handle the user's current utterance, but is unable to take the conversation forward (see Table 3.1, Turn 10) or when the responding RG has finished talking about one topic, and another RG is needed to supply a change of topic (see Table 3.1, Turn 7). The response-and-prompt system makes it easy to always supply the user with a strong path forward in the conversation (e.g. by asking the user a question).

Response Priority Ranking System

We use a priority system to decide which response generator's response should be selected on each turn. When generating responses, each RG provides one of the **response priorities** in Table 3.2.[2] This hierarchy supports the ability to preserve conversational continuity (*STRONG_CONTINUE*), while remaining responsive to the user's initiative (*FORCE_START*). Though it is a relatively simple rule-based system, we have found it well-suited to our needs. The priority levels are clear to understand, and make it easy to modify behavior. By avoiding a centralized response-choosing module, our design allows RGs to decide themselves whether they should respond, and whether their response is high quality. This makes it easier for multiple people to work on different RGs, each with self-contained logic. Lastly, if one RG encounters an error, timeout, or inability to find relevant content, the other RGs provide alternatives.

[2] In case of a tie, we break it using a manually-specified priority ordering of the RGs.

Prompt Priority	Meaning
FORCE_START	This RG should take control. This is mainly used for scripted transitions (Table 3.1, Turn 2).
CURRENT_TOPIC	This RG has a prompt that talks about the current entity (see Section 3.2 and Table 3.1, Turn 9).
CONTEXTUAL	This RG has a prompt that does not talk about the current entity, but that is conditioned on the conversation history, e.g. referring to a previous topic (Table 3.1, Turn 11).
GENERIC	This RG has a prompt that is not conditioned on the conversation so far (Table 3.1, Turn 7).

Table 3.3: Prompt Priorities

Prompt Priority Sampling System

While we use a deterministic ranking system to choose the highest-priority response (Section 3.4.3), *prompts* often represent changes of topic, which are less restricted by context, and (in human-human conversations) tend to have a degree of randomness. Thus, we use a priority *sampling* system to select a prompt. When generating prompts, each RG supplies one of the **prompt priorities** in Table 3.3.

Under the Priority Sampling module, if a *FORCE_START* prompt is supplied, we choose it. Otherwise, we sample from a manually-specified distribution over the remaining priorities, masking out any that are not present on this turn. The distribution is biased towards maintaining continuity of discussion (*CURRENT_TOPIC* $\gg$ *CONTEXTUAL* $>$ *GENERIC*). Then, among the RGs that produced a prompt of the sampled priority, we sample one prompt, using a manually specified distribution over the RGs. This system allows us to specify scripted transitions when desired, and to provide variety via randomness, while still enabling us to tune the likelihood of changing topic, which is an important controllable parameter in chit-chat conversations (See et al., 2019).

3.5 NLP Pipeline

The NLP Pipeline is run at the start of every turn (see Figure 3.1), and contains modules that annotate the user's utterance with information that is useful for other parts of the bot.

3.5.1 CoreNLP

On each turn of the conversation, we annotate the the user's utterance using the Stanford CoreNLP toolkit (Manning et al., 2014), which runs on a remote EC2 module with CPU only. We use the following CoreNLP annotators: tokenization, sentence splitting, part-of-speech tagging, lemmatization, named entity recognition, constituency parsing, dependency parsing, coreference resolution, and sentiment analysis. Due to the format

Training Regime	**# MIDAS Training Set**	**Chirpy Training Set # Silver**	**# Gold**	**Chirpy Test Set Micro-F1**
MIDAS (baseline)	10,090	0	0	0.53
MIDAS+self-training ($\tau = 0.95$)	10,090	41,152	0	0.54
MIDAS+self-training ($\tau = 0.75$)	10,090	62,150	0	0.54
MIDAS+supervised	10,090	0	2,407	**0.81**

Table 3.4: Performance of our Dialogue Act model under different training regimes.

of the user utterances (lowercase with no punctuation), we use the caseless models[3] for part-of-speech tagging, constituency parsing and named entity recognition.

3.5.2 Dialogue Act Classifier

Dialogue acts can support understanding of user intent (Stolcke et al., 2000), and have been successfully employed in previous Alexa Prize socialbots (Yu et al., 2019). To build a dialogue act classifier, we finetuned the HuggingFace implementation (Wolf et al., 2019a) of a BERT-based classification model (Devlin et al., 2019) on the MIDAS dataset (Yu and Yu, 2019). The dataset contains 12,894 examples, where each example is a bot utterance,[4] the user's response to that utterance, and the user's dialogue act.[5] The dataset was collected by Gunrock (Yu et al., 2019), the winner of the 2018 Alexa Prize competition. Unlike other dialogue act datasets, such as SWBD-DAMSL (Jurafsky et al., 1997), which are designed for human-human dialogue, the MIDAS annotation schema was specifically designed for human-chatbot dialogue.

Though this baseline model achieved a micro-average F1-score of 0.78 on the MIDAS test set, we wished to evaluate its performance in our *own* bot's conversational setting. We hand-labeled a 'Chirpy' test set containing 602 examples from our bot's conversations. The same baseline model achieved only 0.53 on this test set (see Table 3.4). We suspect the performance drop is due to the distributional difference between the utterances generated by our bot and by Gunrock. To improve performance on our data, we experimented with self-training (McClosky et al., 2006). Using the baseline model, we labeled many unlabeled examples from our own bot's conversations. Examples whose label was predicted with a confidence score greater than a threshold τ were added to our training set. Using $\tau = 0.75$ and $\tau = 0.95$ added 62,150 and 42,152 silver-labeled training examples, respectively. After training on these expanded datasets, we re-evaluated on our own test set.

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

[4] The bot utterance is included because it contains context essential to understand the user utterance (Yu and Yu, 2019). For instance, the user utterance 'tiger king' is an *opinion* when in response to 'What is the best show?' and a *statement* when in response to 'What is the last show you watched?'.

[5] To better fit our needs, we modified the label space as described in Section A.3.1.

The inclusion of the silver-labeled data did not substantially boost performance (see Table 3.4). Finally, we turned to supervised training, and hand-labeled an additional 2,407 examples from our own bot's conversations (procedure described in Section A.3.2). After training on the MIDAS data and this data, we achieved a much higher micro-F1 of 0.81 on the Chirpy test set.

In our bot, we run the Dialogue Act classifier on an EC2 machine with one NVIDIA T4 Tensor Core GPU, annotating every user utterance in the conversation. We find that its accuracy is best on classes with low variance in user utterances, such as *positive answer*, while classes with high variance, such as *statement*, are more difficult. However, even for the low variance classes, the classifier's labels are very useful – we are able to achieve much higher recall in recognizing *positive answer* and *negative answer* by using the classifier's labels, compared to regexes or word lists.

3.5.3 Question Classifier

Users often spontaneously ask factual questions, personal questions, follow-up questions, and even questions unrelated to the current topic. Recognizing and answering these questions is important, particularly for user initiative, but is also non-trivial, as user utterances do not contain punctuation.

To recognize questions, we initially used the Dialogue Act classifier's labels (which include question types like *factual question* and *open-ended question*). However, this did not work well; the classifier seemed to condition too much on the bot utterance preceding the user utterance – which is less useful for recognizing questions than other dialogue acts. Instead, we fine-tuned a RoBERTa model (Liu et al., 2019; Wolf et al., 2019a) on a simplified version of the Dialogue Act training data, framing the task as binary classification, conditioned only on the user utterance. This model achieved an F1-score of 0.92 and improved the reliability of question detection.

The classifier's labels are used to determine when certain RGs should respond – for example, when the Evi RG (Section A.1.3) should answer a factual question. The labels are also useful for the neural generative models (Sections 3.7, 3.8, 3.6.5). We observe that the GPT-2-based models are much more likely to answer (rather than ignore) a user's question if a question mark is present. Thus, we use the classifier labels to determine when to append a question mark to the user utterance.

3.5.4 Entity Linker

A key part of our high-coverage strategy (Section 3.1) is *entity linking* – detecting when the user is referring to an entity, and identifying the correct entity. To obtain our pool of potential entities, we processed a dump[6] of

[6] https://dumps.wikimedia.org

English language Wikipedia. For each article (i.e. each entity E), we collected (a) the *pageview* (number of views in one month), and (b) the *anchortext distribution* $P_{\text{anchortext}}(a|E)$.

To compute the anchortext distribution for an entity E, we count the number of *anchortexts* (i.e., strings, lowercased) that are used as hyperlinks to E across Wikipedia (e.g., the entity Barack Obama may be referred to using the anchortexts *Barack Obama*, *obama*, or *president obama*). Then:

$$P_{\text{anchortext}}(a|E) = \frac{\text{count(links from } a \text{ to } E)}{\sum_{a' \in A(E)} \text{count(links from } a' \text{ to } E)} \tag{3.1}$$

where $A(E)$ is the set of all anchortexts that link to E. We store each entity, along with its Wikipedia article, pageview, anchortext distribution, and Wikidata categories[7] in an ElasticSearch index.

After we receive the user's utterance u, we assemble the set of candidate spans S. S contains all n-grams in u with $n \leq 5$, excluding n-grams that consist only of stopwords. We then query ElasticSearch to fetch all entities E which have at least one span $s \in S$ among its anchortexts. To determine which entities the user is referring to, we wish to estimate $P(E|s)$, the likelihood that a span s is referring to an entity E. We model $P(E|s)$ as a Bayesian system:

$$P(E|s) \propto P(E) \times P(s|E). \tag{3.2}$$

We assume that $P(E)$ is proportional to the pageview for the entity E, and $P(s|E) = P_{\text{anchortext}}(s|E)$. Therefore, we define the score(s, E) of a span s and entity E to be:

$$\text{score}(s, E) = \text{pageview}(E) \times P_{\text{anchortext}}(s|E). \tag{3.3}$$

The output of the entity linker is a priority-ordered list of (s, E) pairs. The ordering is calculated using manually-curated rules and thresholds on the following features: (a) the score of (s, E), (b) the maximum unigram frequency[8] of s, (d) whether E is in a Wikidata category that is expected for this turn[9], (c) whether s is contained inside any other linked span (priority is usually given to the larger span). The output of the entity linker is primarily used by the entity tracker (Section 3.4.2) to identify the current entity under discussion.

Limitations We found the entity linker to be one of the hardest components of our bot to build. One difficulty is that our notion of an entity – anything with a Wikipedia article (e.g. *Cat* or *Musical instrument* in Table 3.1) – is much broader than the traditional definition of Named Entities (which is typically restricted to

[7]For each entity, we collected all its ancestors via the *instance of* and *subclass of* relations. For people entities, we also used the *occupation* relation.

[8]The maximum unigram frequency of s is the frequency of the most common unigram inside s, computed using this unigram frequency list for spoken English: http://ucrel.lancs.ac.uk/bncfreq/flists.html

[9]For example, if the bot asked *What's your favorite movie?*, an expected Wikidata category is *film*.

particular types, such as people and locations). Our motivation in this definition was to enable high-coverage world knowledge by enabling any Wikipedia article to become a focus of discussion. However, this made the entity linker's job much more difficult. The need to detect an extremely broad range of entities, with no restriction to certain types, made it much more difficult to find a good precision/recall tradeoff, leading to both false positive and false negative problems in the bot. In Chapter 5, I train a neural retriever that bypasses this step of heuristic entity linking. Instead, it relies on contextual embeddings to find mentions of entities in text-passages from Wikipedia. With some careful modifications, a high-quality entity linker can be derived from it.

ASR Error Robustness As we do not have access to original user audio, ASR errors are a major source of difficulty, particularly when they occur within entity names. For example, if the user wants to talk about the film *Ford v Ferrari*, but the ASR transcription is *four v ferrari*, our entity linker will fail to identify the correct entity, as the span *four v ferrari* is not among the anchortexts for the entity *Ford v Ferarri*. To address this, we adapted our entity linker to be robust to phonetically-similar spans and anchortexts; our method is similar to the method by Chen et al. (2018).

First, we converted all Wikipedia entity anchortexts to their phoneme and metaphone representations (e.g., *Harry Potter* to `'HH EH R IY P AA T ER'` and `'HRPTR'`) with a grapheme-to-phoneme tool[10] and the double metaphone algorithm,[11] and indexed the mapping from anchortext phonemes to Wikipedia entities in ElasticSearch. When running the entity linker, we convert all spans $s \in S$ to their phonetic representations and query the ElasticSearch index, which returns a set of anchortexts A_{phon} that have similar phonetic representations to any of the spans queried. This allows us to expand the candidate pool for each span s, from entities for which s *is* an anchortext, to entities for which s is *phonetically similar* to an anchortext. Finally, we redefine $P(s|E)$ as follows: for each anchortext $a \in A_{\mathrm{phon}}$, we start by finding its best-matching span $s^*(a) = \arg\max_{s \in S} \mathrm{sim}(s, a)$ where $\mathrm{sim}(\cdot, \cdot)$ is a phoneme similarity function[12] between 0 and 1; then, we filter out anchortexts that are phonetically too dissimilar to each span with a threshold of 0.8, resulting in a set of anchortexts for each span $A(s) = \{a | a \in A_{\mathrm{phon}}, s = s^*(a), \mathrm{sim}(a, s) \geq 0.8\}$. Finally:

$$P(s|E) \propto \begin{cases} \max_{a \in A(s)} \text{count(links from } a \text{ to } E) \times \mathrm{sim}(s, a) & A(s) \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

This definition of $P(s|E)$ replaces $P_{\text{anchortext}}(s|E)$ in Equation (3.3).

[10] https://pypi.org/project/g2p-en/
[11] https://pypi.org/project/metaphone/
[12] implemented on lists of phonemes with Python's `difflib.SequenceMatcher`

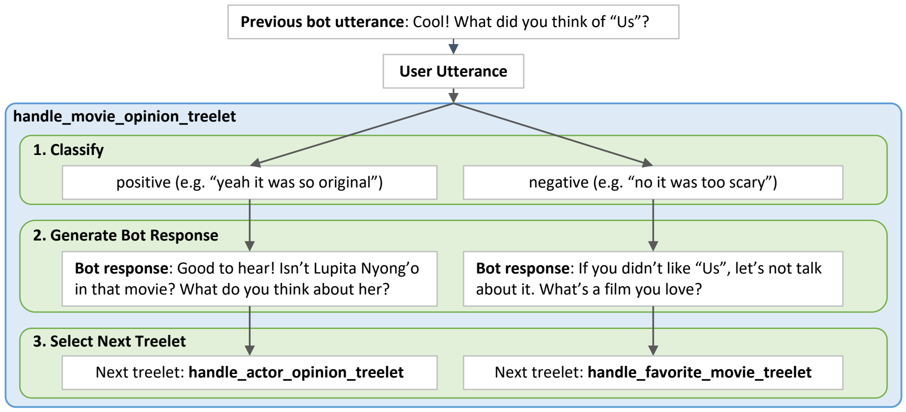


Figure 3.2: An example *treelet* for the Movies RG.

3.6 Response Generators

In this section, we describe our Response Generators (RGs). Additional minor RGs are described in Appendix A.1. We also describe *treelets* (Section 3.6.1), a system we used to organize many of our RGs.

3.6.1 Treelets: A System to Organize Dialogue Graphs

Many of our response generators rely on *treelets*, a modular programming abstraction which represents a single node in a dialogue graph. The treelet system is inspired by simple rewriting rules from ELIZA (Weizenbaum, 1966) and dialogue-frame-based systems such as GUS (Bobrow et al., 1977). We define a treelet to be a small, 1-turn dialogue 'tree' that manages all decisions necessary to produce a bot response given a user's utterance. This involves interpreting the user utterance, creating the bot's response, and specifying the treelet that should take control on the next turn.

Typically, a treelet performs three actions: (1) it classifies the user's utterance into one of several branches, (2) it produces an appropriate bot response for that branch, (3) it specifies the next treelet. Treelets throughout our bot may classify user utterances by using regexes, outputs from our NLP pipeline (the dialogue act classifier is frequently used for this purpose), or changes in entity (e.g., if a treelet in the Movies RG detects that the current entity has changed to "food" after the user says "let's talk about food", the current Movies treelet may select a branch that returns no response). Bot responses may be handwritten or dynamically generated (we use both throughout our system). An example from the Movies RG is shown in Figure 3.2.

Like dialogue trees in general, treelets provide a well-controlled, predictable and easily interpretable conversation flow. From an engineering and implementation perspective, treelets have several advantages, such as allowing modular organization of code and dialogue, easily enabling cycles when desired (by having treelets point to each other with repeats or loops), and minimizing code duplication by allowing many treelets to point to the same successor.

3.6.2 Opinion

Exchanging opinions is a core part of social chit-chat. To form a stronger sense of personality, and to seem more relatable, it is important that our bot can also express its opinions. The Opinion RG's goal is to listen to users' opinions on certain topics, and reciprocate with its 'own' opinions (sourced from Twitter) on those topics.

Data To collect both positive and negative opinions, we queried a Twitter stream[13] using a regex to collect tweets of the form `'i (love|like|admire|adore|hate|don't like|dislike) TOPIC because REASON'`, where `TOPIC` and `REASON` can be any text. We collected 900,000 tweets, which are stored in a Postgres table hosted on AWS Relational Database Service (RDS). Of these, we manually whitelisted 1012 reasons across 109 popular topics. To avoid speaking inappropriately about sensitive topics, we only whitelist uncontroversial entities (such as animals, foods, books/movies/games, everyday experiences such as working from home, being sick, days of the week, etc.), and ensured that all reasons, including negative ones, are inoffensive and good-spirited.

Behavior Currently, the Opinion RG activates when the user mentions one of the whitelisted entities (e.g. Table 3.1, Turn 8). We ask whether the user likes the entity and classify their response using the CoreNLP sentiment classifier (Section 3.5.1). We then either agree or disagree with the user. If we disagree, we either ask the user for their reason for their opinion, or supply a reason why we disagree, and ask what they think of our reason. Ultimately, we want the user to have a positive experience with our bot, so regardless of whether we disagree or agree with the user, we will ask the user their opinion on a related entity, and always agree with the user about the new entity. The conversation may end earlier, as we detect on each turn whether the user is still interested via their utterance length. If the utterance contains less than 4 words, and it does not contain any of the 'agreement' words (such as 'same', 'me too', etc.) we will hand off the conversation to another RG. Even when the RG is not active, it keeps track of whether the user has already expressed an opinion on an entity, by applying a regex similar to that applied to the tweets.

Agreement Policies Disagreement is an unavoidable part of human-human conversations, and we hypothesize that occasional disagreement is necessary in order for our bot to have a convincing and individual personality. To test this, we implemented three policies (full details in Section A.6): (i) `ALWAYS_AGREE`

[13] https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data

Policy Name	Continuation Rate (%)	CI (%)
CONVINCED_AGREE	52.7	3.5
ALWAYS_AGREE	58.7	0.9
LISTEN_FIRST_DISAGREE	58.7	1.3

Table 3.5: **Continuation rate for each agreement policy.** The Confidence Intervals (CI) differ due to different sample sizes (ALWAYS_AGREE receives 0.5 of traffic, LISTEN_FIRST_DISAGREE receives 0.3, CONVINCED_AGREE receives 0.2).

– we always agree with the user's sentiment on the entity; (ii) LISTEN_FIRST_DISAGREE – first we ask the user's reason for liking/disliking the entity, then we offer our reason for disagreeing with their sentiment; and (iii) CONVINCED_AGREE – we initially disagree with the user's sentiment on the entity, but after the user gives their reason for liking/disliking the entity, we switch our sentiment to match the user's (i.e. we are convinced by the user). To evaluate the policies, we ask the user *Would you like to continue sharing opinions?* and interpret the desire to continue is an indication of a successful policy. Table 3.5 shows that users prefer ALWAYS_AGREE and LISTEN_FIRST_DISAGREE over CONVINCED_AGREE, and all policies have high continuation rates, suggesting that disagreement can be a positive and stimulating part of a conversation, but that the manner and delivery of the disagreement is an important factor.

3.6.3 Movies

The Movies RG is designed to deliver a high-quality scripted conversation about a movie the user specifies, using information drawn from the Alexa Knowledge Graph.[14] Currently, the RG is activated when the user asks to talk about movies, mentions a movie keyword (such as *movies* or *film*) or talks about any movie-related entity (e.g. *Saving Private Ryan*, *Meryl Streep*, *the Coen brothers*, etc.). Once activated, the RG typically asks the user to name a movie, asks the user's opinion on it, gives a fun fact about the movie, asks the user their opinion on an actor in the movie, then asks the user if they've seen a different movie featuring that actor (see Turns 4-7 in Table 3.1). The RG uses treelets (Section 3.6.1) to organize the dialogue graph, handwritten templates to form the bot utterances, and a mixture of regexes and the CoreNLP sentiment classifier (Section 3.5.1) to classify the user's responses.

The primary weakness of this RG is that, as a scripted dialogue graph, it does not offer very high user initiative (one of our design goals – Section 3.1). However, this RG was important especially early in the competition when our more flexible RGs were still under development, and we needed more content. Another difficulty we faced was the latency of the Alexa Knowledge Graph, which was sufficiently slow that we were

[14] The Alexa Knowledge Graph is an Amazon-internal resource; our team was given access to parts of it.

limited to one query per turn; this limited the scope of interesting information that we could pull about an entity and heavily influenced the design of our dialogue tree.

3.6.4 Music

Similar to the Movies RG, the Music RG is designed to deliver scripted conversations about musical entities that the user specifies. The RG is activated when a musician/band or a music keyword (such as *music* or *songs*) is mentioned. Once activated, the Music RG engages in a conversation specific to the type of the musical entity that was mentioned. Unlike the Movies RG, the Music RG has a randomized internal prompting system that allows the conversation to be centered around music even when a scripted conversation is exhausted for a specific entity. For example, after the Music RG goes until the end of a scripted conversation for a musician, it can ask for an internal prompt, and start a conversation about musical instruments, songs, or music in general. The randomized nature of the internal prompting system makes the conversation more flexible, and mitigates some weaknesses of scripted conversations mentioned in Section 3.6.3.

3.6.5 Neural Fallback

Our Fallback RG's responses – e.g., *Sorry, I'm not sure how to answer that* (Section A.1.3) – are a poor user experience, making the user feel ignored and not understood. The Neural Fallback RG aims to generate a better fallback response using our GPT-2 EmpatheticDialogues model (Section 3.7) – to be used only if every other RG (excluding Fallback) has no response. If the neural fallback response is chosen, another RG immediately produces a prompt to move the conversation in another direction. After some filtering (e.g. removing responses that ask questions or give advice), the neural fallbacks can work well as a way to better acknowledge and show understanding of what the user said, such as on Turn 11 of Table 3.1. A remaining issue is latency – generating from the GPT-2 model is typically the slowest component in the turn, which is a poor tradeoff if we don't use the neural fallback.

3.6.6 Categories

The Categories RG was originally designed to ask handwritten questions about certain categories; for example, *Where's a place you would love to visit?* for the 'travel' category. These questions may be asked when the current topic is 'travel', or used as generic changes of topic (Table 3.1, Turn 7). The goal is for the user to name an entity (e.g. *Japan*) that can form the basis for an interesting discussion (e.g. with the Wiki or Opinion RGs). However, we found that repeatedly asking users to think of entities led to decision fatigue, with many

Strategy	Percentage of Turns with New User Entities	CI (%)
STATEMENT	27.2	1.2
QUESTION	26.4	2.7
STATEMENT+QUESTION	32.8	1.6

Table 3.6: **Rate at which users suggest new entities, for different strategies in the Categories RG.** The entities are extracted using our Entity Linker (see Section 3.5.4). (CI: Confidence Interval)

Strategy	Re-offense Rate (%)	Confidence Interval (%)
WHY	52.0	±4.9
WHY+NAME	*63.8*	±*7.0*
AVOIDANCE	55.4	±4.9
AVOIDANCE+NAME	39.1	±6.1
AVOIDANCE+PROMPT	58.3	±4.7
AVOIDANCE+NAME+PROMPT	**34.6**	±**6.6**
COUNTER+PROMPT	56.7	±4.2
EMPATHETIC+PROMPT	46.1	±4.6

Table 3.7: **Re-offense rates for different response strategies to offensive utterances.** Italic and bold denote the worst and best performing, respectively.

users failing to think of an entity.[15] As alternatives to the QUESTION strategy, we experimented with two other strategies: STATEMENT, in which the bot just makes an observation about a relevant entity (e.g. *Mexico is one of my favorite places. I love the food and beaches!*), and STATEMENT+QUESTION, which combines the other two strategies. Table 3.6 shows that the statement followed by a question elicited the most new entities. This may be because the statement gives users an example, and takes the focus off the user for a moment, before prompting them with a question. This is a more natural, mixed-initiative experience than simply asking a question.

3.6.7 Offensive User

Users sometimes give offensive or critical utterances, and it is important for our bot to handle these appropriately (Curry and Rieser, 2018, 2019). Unsurprisingly, there is an inverse relationship between the presence of offensive user utterances in a conversation and the conversation rating (Figure 3.8). Our goal is to redirect the user away from making offensive comments, towards topics the bot can discuss.

On each turn, the Offensive User RG checks the user's utterance for offensive language using a blacklist of

[15] If the user does not name a new entity, we respond either with a handwritten acknowledgement and new question (if the user said *I don't know* or similar), or with the GPT-2 model (Section 3.6.5).

offensive phrases.[16] If the user's utterance is more critical than offensive, we respond with an apologetic strategy (see Turn 10 of Table 3.1). For offensive user utterances, we implemented two immediate response strategies: asking the user why they made the offensive remark (`WHY`); or politely avoiding the topic (`AVOIDANCE`). In addition, for `AVOIDANCE`, we experimented immediately changing the topic by using a prompt in the same turn (`AVOIDANCE+PROMPT`). For each of these configurations, we experimented with mentioning the user's name (`NAME`), or not. We also implemented the strategy `COUNTER+PROMPT`, inspired by Brahnam (2005), which directly confronts the user before changing topic, and `EMPATHETIC+PROMPT`, inspired by Chin et al. (2020), which empathizes with the user before changing topic. The full details can be found in Appendix A.5 and also in a follow up work by Li et al. (2021).

Table 3.7 shows the effect of each strategy on re-offense rate (i.e., the probability that the user says another offensive utterance in the same conversation). We find that mentioning the user's name reduces the likelihood of re-offense when we use the avoidance strategy, but increases re-offense rate when we ask the user why they made an offensive remark. We hypothesize that by using their name, we motivate the user to defend themselves, which prolongs the offensive conversation. We find that our `AVOIDANCE+NAME+PROMPT` method outperforms the empathetic method (`EMPATHETIC+PROMPT`) and the confrontation method (`COUNTER+PROMPT`).

3.7 Neural Chat

The Neural Chat RG's goal is to empathetically discuss personal experiences and emotions with the user, using responses generated by a GPT-2-medium (Radford et al., 2019) model fine-tuned on the EmpatheticDialogues dataset (Rashkin et al., 2019). The dataset consists of conversations between a *speaker*, who describes an emotional personal experience, and a *listener*, who responds empathetically to the speaker's story. Our model is trained in the listener role.

The Neural Chat RG has 7 discussion areas: current and recent activities, future activities, general activities, emotions, family members, living situation, and food. A discussion begins by asking the user a **starter question** (e.g, *What do you like to do to relax?* for the 'general activities' area). Some starter questions are conditioned on the time of day (e.g. *What did you have for breakfast/lunch/dinner today?* for the 'food' area). Starter questions can be asked as part of the launch sequence (Table 3.1, Turns 2 and 3), as generic changes of topic, (*Do you have any plans for the weekend?*), or can be triggered contextually (*You mentioned your boyfriend. How did you guys meet?*). On each subsequent turn of the discussion, we generate 20 possible responses from the GPT-2 model using top-p sampling with $p = 0.9$ and temperature 0.7. If over a third of

[16] `https://www.freewebheaders.com/full-list-of-bad-words-banned-by-google/`. Our offensive classifier is also used by our RGs to check that externally-sourced content (e.g. news articles, Wikipedia articles, fun facts) are inoffensive.

Strategy	Preamble
NO_SHARE	I wanted to check in with you.
POS_OTHERS	I've noticed that a lot of people are feeling pretty positive today!
POS_BOT	I wanted to say that I'm feeling pretty positive today!
POS_BOT_STORY	**POS_BOT** + I just went for a walk outside, and it felt great to get some fresh air.
NEG_OTHERS	I've noticed that a lot of people are feeling kind of down recently.
NEG_BOT	I wanted to say that I've been feeling kind of down recently.
NEG_BOT_STORY	**NEG_BOT** + I've been missing my friends a lot and finding it hard to focus.
NEGOPT_OTHERS	**NEG_OTHERS** + But I think it's important to remember that things will get better.
NEGOPT_BOT	**NEG_BOT** + But I think it's important to remember that things will get better.
NEGOPT_BOT_STORY	**NEGOPT_BOT** + Just earlier today I took a walk outside and the fresh air helped me get some perspective.

Table 3.8: **Strategies for the emotion-focused Neural Chat starter question. POS/NEG/NEGOPT** refer to positive/negative/negative+optimistic emotion. **OTHERS/BOT** refer to whether the emotion is attributed to other people, or to the bot. **STORY** indicates that the bot shares a personal anecdote.

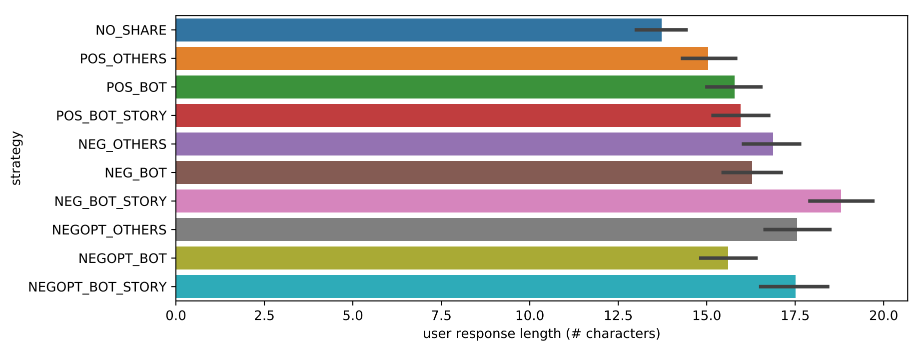


Figure 3.3: Effect of Neural Chat emotion-focused starter question strategies on user response length.

the sampled responses contain questions, we choose one of them to provide a strong path forwards in the conversation. If not, we interpret this as an indication that the model is not confident in asking a question on this turn. In this case, we choose a non-question and end the Neural Chat discussion. Under this strategy, each Neural Chat discussion contains 2.75 bot utterances on average.

The model was fine-tuned using the HuggingFace ConvAI code[17] (Wolf et al., 2019b) and is hosted on a GPU-enabled EC2 machine with one NVIDIA T4 Tensor Core GPU. To keep latency low we truncate the conversational history supplied to the model, so that the total number of GPT-2 tokens is below 800. Given that neural models have been shown to make poor use of longer conversational history (Sankar et al., 2019), this truncation does not seem to be a limiting problem currently.

[17] https://github.com/huggingface/transfer-learning-conv-ai

Emotion-focused Conversations As part of our goal to provide an emotionally-engaging experience (Section 3.1), we would like to give users space to share their genuine feelings, then respond empathetically to them. This was especially important during the Coronavirus pandemic (Appendix A.1.1), which was an emotionally challenging time for many. Given our basic starter question *I hope you don't mind me asking, how are you feeling?*, we tried several preambles to precede the question (Table 3.8). Figure 3.3 shows the effect of the different strategies on the length of the user's response. We find that the basic **NO_SHARE** strategy has the shortest average response length, indicating that the bot's emotional observations (whether about the bot or about other people) lead users to give more substantive responses. Users tend to give longer responses when the bot expresses negative emotions (**NEG** and **NEGOPT**) than positive (**POS**) – this may be because acknowledging negative emotions makes users feel more comfortable to answer the question honestly, rather than superficially (e.g. *i'm fine*). Furthermore, adding a personal anecdote (**STORY**) to the negative bot emotions led to longer responses – users may have responded more because the bot was more specific or relatable. For positive emotions (**POS**), users are more responsive when the bot attributes the positive emotion to itself (**BOT**), than to other people (**OTHERS**). However, for negative emotions (**NEG** and **NEGOPT**), the opposite is true. We also experimented with including the user's name in the starter question, but found that this made no difference to user response length.

Discussion Our neural generative model has several recurring weaknesses which impact overall user experience. First, it frequently asks for already-provided information, asks non sequitur questions, makes unfounded assumptions about the user, and confuses its own previous responses with the user's. This demonstrates that incorporating commonsense reasoning is a priority in neural generation. Second, while the model generally produces interesting and relevant responses to longer user utterances, it performs poorly when the user utterance is short or low-content (e.g. *okay*, *i don't know*, *nothing*) – probably because these utterances are unlike the much longer and meaningful EmpatheticDialogues training data. The model tends to respond to these with bland responses that further fail to drive the conversation to any interesting substance. This problem with short user responses is one reason why we focused on finding starter questions that lead to substantial user responses (Figure 3.3).

Due to these difficulties, most conversations with the GPT-2 model tend to fall apart after a few turns, as the bot will eventually ask a question that doesn't make sense, which will flummox the user. This is one reason why we designed the Neural Chat module around shorter sub-conversations. However, overall, we are excited that neural generation is now able to interact successfully with real people, within certain constraints (such as keeping the discussion short, bookending it between handwritten starter questions and wrap up phrases, and providing a strong path forward through questions).

3.8 Wiki Response Generator

To support our goal of high-coverage world knowledge (Section 3.1), the Wiki RG uses Wikipedia articles as grounding to discuss any entity that interests the user. Our goal is to allow the user to conversationally discover interesting information about the entity.

Data To prepare the Wikipedia data, we downloaded the most recent Wikipedia dump,[18] processed it using MWParserFromHell[19] and Spark,[20] and uploaded it into an ElasticSearch index. The Wiki RG can then query the ElasticSearch index to obtain the Wikipedia article for an entity.

Behavior On each turn, if it's not already active, the Wiki RG can start to talk about the current entity (Section 3.4.2) by asking the user an **open-ended question**, such as *What do you find interesting about it?*. If the entity is in one of 25 commonly-encountered types (determined using Wikidata categories), such as books or foods, we use a more specific question, such as *What did you think of BOOK_ENTITY's story?* or *I love trying out new flavor combinations. What do you like to have FOOD_ENTITY with?*. These questions are designed to elicit contentful user responses, which can be matched to specific sentences in the Wikipedia article using TF-IDF overlap. The RG also offers interesting facts (i.e. 'TILs') scraped from the /r/todayilearned subreddit, if available. If we have given enough TILs or we have no TIL left to offer, we will start suggesting sections of the Wikipedia article to the user. A short example Wiki interaction is shown in Turns 11-13 of Table 3.1.

Conversational Styling We use this RG as a testbed for our conversational paraphrasing system. The system takes as input the truncated conversational history, and some knowledge context (either a TIL about the current entity, or an excerpt of the Wikipedia article, selected based on TF-IDF similarity to the user's response to an open-ended question). It outputs a conversational-sounding paraphrase of the knowledge context. The model was trained by fine-tuning a GPT-2-medium language model (Radford et al., 2019) on a processed and filtered version of the TopicalChat dataset (Gopalakrishnan et al., 2019). The paraphrases are generated using top-p decoding with $p = 0.75$ and temperature $\tau = 0.9$, and we pick the one which has the highest unigram overlap with the knowledge context.

Challenges One major challenge while performing conversational styling is that the model sometimes produces **factually incorrect** or nonsensical conversational paraphrases. Another challenge is that integrating the paraphrasing model with the rest of the system requires **explicit directives** such as "continue talking about same knowledge piece", "pick another fact", "change entity" which the model currently does not produce. For instance, sometimes the generated paraphrase just asks a question or mentions an incomplete piece of

[18] https://dumps.wikimedia.org/backup-index.html
[19] https://mwparserfromhell.readthedocs.io/en/latest
[20] https://spark.apache.org

information, with the expectation of completing it in the next turn. Currently, we apply some heuristics such as presence of *Did you know ... ?* style questions or low unigram overlap to determine that the same snippet needs to be paraphrased again.

More broadly, there are challenges around **interestingness of content**. The majority of content on Wikipedia isn't very interesting and social. While the TILs remedy that to some extent, finding interesting parts of raw text is still an open question and quite important in the open-domain conversational setting. Another major challenge is **content selection and discoverability**. The user doesn't know the extent of the knowledge that our system possesses for an entity. In a visual interface, the user can scroll through the article or look at a table of contents. While we partly remedy this by suggesting section titles to illustrate the kind of content we can talk about, a better system could perhaps understand what different parts of a Wikipedia article are talking about, and steer conversation in that direction.

3.9 Analysis

3.9.1 Relationship between Rating and Engagement

We measured four metrics of engagement: number of turns in the conversation, number of distinct entities discussed during the conversation, average length of the user's utterances, and average length of the bot's utterances. Figure 3.4 shows that rating increases with number of turns and number of entities, but ultimately drops off. In an analysis of Alexa Prize bots, Venkatesh et al. (2018) found that across all bots, conversation length was positively correlated with rating; however, one possible explanation for our result is that our bot has limited content and at some point, the users become dissatisfied as their experience is no longer novel.

In an analysis of the NeurIPS ConvAI2 challenge, Dinan et al. (2019a) found a positive relationship between user utterance length and rating. We expected a similar result, thinking more talkative users would be more actively engaged. However, Figure 3.4 shows that rating increases with user utterance length until

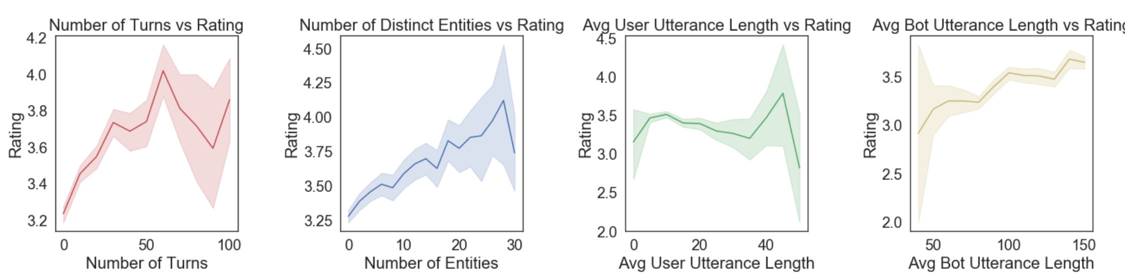


Figure 3.4: Engagement metrics vs rating

about 12 characters, and then decreases. Since many of our bot's questions encourage short answers (e.g. *What's your favorite animal?*; *Would you like to talk about science?*), and it is generally more difficult for our bot to correctly understand and handle longer answers,[21] users who give longer answers may have a worse experience. For this reason, the result shown may reflect the limitations of our bot, more than a user preference for giving shorter responses.

Average bot utterance length is positively correlated with average rating, with high variance in rating for shorter bot utterances. A confounding factor is that different response generators have varying average response lengths and relationship with user experience (Section 3.9.4) – e.g., the Offensive User RG tends to give short responses, and has a negative relationship with ratings. Response generators giving longer responses tend to have positive or neutral relationships with rating. Therefore, this plot may more reflect the UX of our response generators than a user preference for longer responses. These results may also reflect the inherent noise in user Likert-scale ratings (Liang et al., 2020).

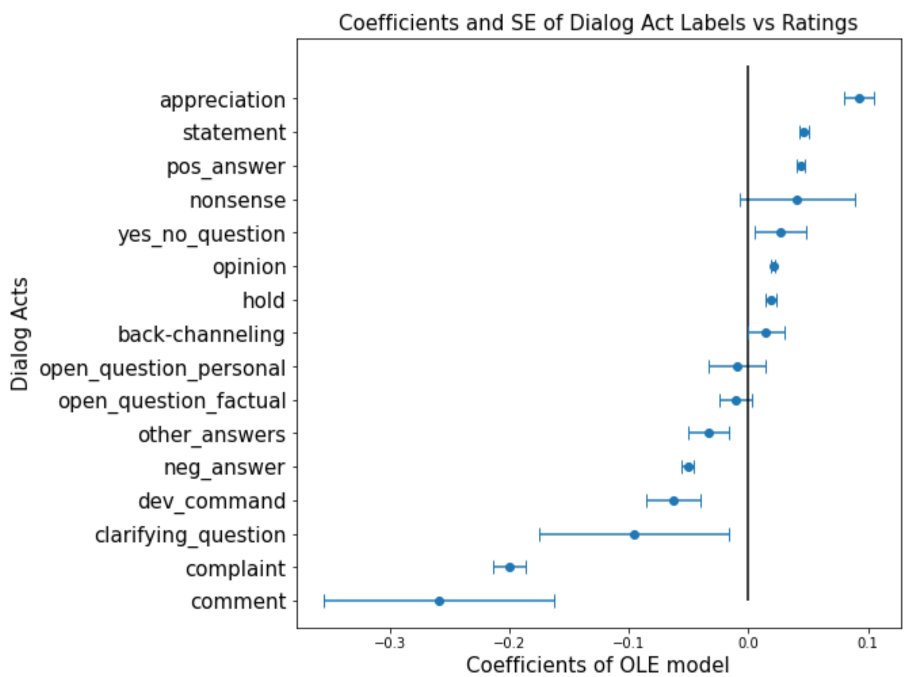


Figure 3.5: Regression coefficients for Dialogue Act vs Rating

[21] As an exception, our neural generation models perform *better* on longer user utterances; see Section 3.7.

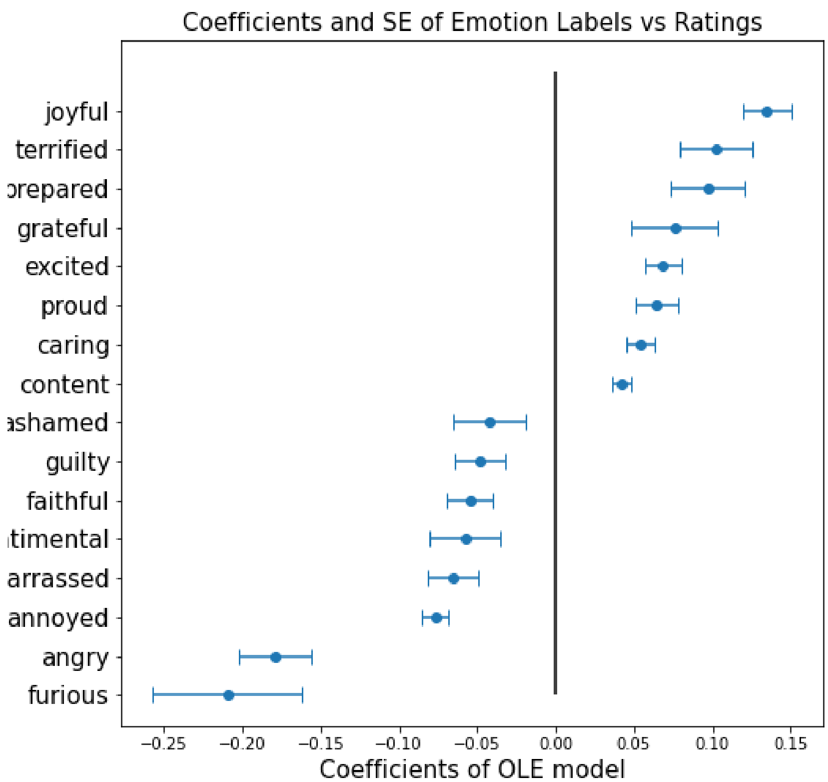


Figure 3.6: Regression coefficients for Emotion vs Rating

3.9.2 Relationship between Rating and User Dialogue Acts

To understand how users' dialogue acts relate to our bot's performance, we applied a regression analysis, using the statsmodels (Seabold and Perktold, 2010) implementation of Ordinary Least Squares, to the distinct dialogue act classifier labels for all utterances of a conversation and the ultimate rating of that conversation. These results are shown in Figure 3.5. As we would expect, *appreciation* is associated with higher ratings and *complaint* with lower ratings.

One of our design goals was having mixed-initiative dialogue. In general, dialogue acts associated with low user initiative, such as *comment*, *pos_answer*, *statement*, and *back-channeling* were more positively associated with rating than dialogue acts associated with high user initiative, such as *command*, *open_question_opinion*, and *open_question_factual*. A possible explanation for this is that users take more initiative when dissatisfied with the current conversational direction, for example by giving a command to change the topic. On the other

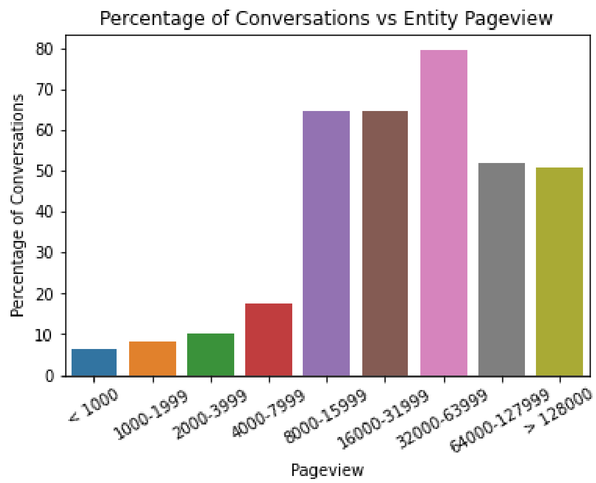


Figure 3.7: Percentage of conversations in which users initiated discussion of entities with different popularity levels (pageview).

hand, users giving yes-answers or back-channeling, are likely being compliant with the bot's direction, which may reflect greater overall satisfaction. It is possible that these results are more indicative of user satisfaction with our content than of a user preference for low vs high initiative.

3.9.3 Entity Coverage

As part of our design goal to offer high coverage of topics (Section 3.1), our bot is capable of discussing any Wikipedia entity (Section 3.4.2), and discussed 7.5 distinct entities on average per conversation. To support user initiative and engage users, we designed our bot to be able to discuss both popular and lesser-known entities. We regard the Wikipedia pageview (Section 3.5.4) as a measure for an entity's popularity. To measure users' desire to discuss less-common entities, Figure 3.7 shows the percentage of conversations where users initiated discussion of an entity with different pageview levels. These counts do not include entities initiated by the bot. As the plot shows, a significant number of users wanted to discuss uncommon entities: in 8% of our conversations, users initiated discussion of entities with fewer than 2000 views and 33% of conversations covered at least one entity with fewer than 8000 views. Users who discussed rare entities with the bot appeared to have favorable experiences. Conversations with rare entities (fewer than 16000 pageviews) had an average rating of 3.88, while those without rare entities had an average rating of 3.64.

To understand which entities had the greatest impact on user experience, we used the top 100 most frequent entities as features for a regression analysis, using an Ordinary Least Squares model. Of the 100 most popular entities, 15 had a statistically significant ($p \leq 0.05$) positive impact on rating. These include **animals** ('Cat',

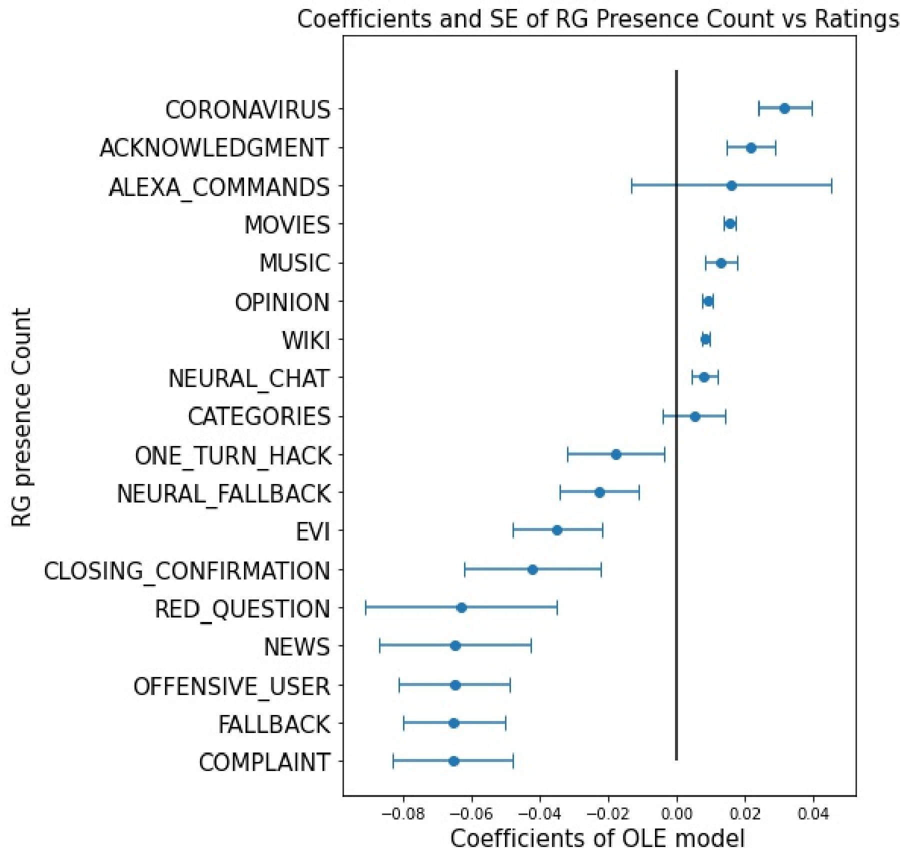


Figure 3.8: **Regression coefficients for Response Generator vs Rating.** Launch RG is not included as it is in every conversation.

'Dog'), **movies** ('Film', 'Frozen 2', 'Onward (film)'), **food** ('Korean fried chicken', 'Pizza', and 'Ice cream'), and **video games** ('Minecraft', 'Fortnite').

3.9.4 Effectiveness of Response Generators

We performed a regression analysis on the relationship between response generator use and rating, using the number of turns each RG contributed as features. Figure 3.8 shows a statistically significant positive relationship between rating and the Coronavirus, Acknowledgement, Movies, Opinion, and Wiki RGs, and a statistically significant negative relationship for Red Question, Complaint, Fallback, Neural Fallback, and Offensive User. The Complaint and Offensive User results may be explained by the fact that users experiencing poor conversations may complain or be offensive, and conversely, some adversarial users deliberately engage negatively and then give poor ratings. A possible cause for the negative Fallback and Neural Fallback results is that these RGs are used when no other RG has a high-quality response, so their use is likely correlated with a worse user experience. As we expected, RGs designed for general conversation had more positive coefficients. Of these RGs, those with more scripted content, i.e. Coronavirus, Acknowledgement, Movies, and Categories, had larger positive coefficients than those with less, such as Opinion and Wiki. However, the most significant loss in performance occurs when the bot cannot answer contextually or has an adversarial user.

3.10 Discussion

Full Stack NLP Most NLP research focuses on self-contained tasks. However, an open-domain socialbot, served to a diverse range of customers in widely different contexts, is by no means a self-contained task. Our socialbot is a tapestry of many such components, requiring a deep understanding of each component and how they should work together – a setting we call Full Stack NLP. Often the inputs and outputs of these components are inter-dependent, leading to cascading errors. We made many design choices which delay hard decisions in pipelines, and maximize information exchange between modules. This calls for research on models which perform these tasks jointly and methods which enable training over multiple interdependent tasks with only a small amount of joint supervision. Within the domain of informative conversations, we address this issue in Chapter 5 where we train a retriever and a generator jointly without any passage supervision.

Domain Shift As a recurring problem, we found that many existing NLP resources didn't work well out-the-box. The main reason for this is that the training data for these resources (typically non-conversational, long form, traditionally-formatted written text) is misaligned with our setting (conversational, short form, uncased, no punctuations, spoken text). However, a deeper reason is the constantly changing nature of dialogue agents themselves. Even for an extremely related resource (the MIDAS dialogue model, developed for the Alexa Prize, Section 3.5.2), domain shift was a problem. Recent advances in online- and meta-learning could provide a useful long term solution to this issue. Further, when a model fails, the system needs to recover. See and Manning (2021) from our team show how to train a model to predict next-turn failure based on prior examples of dissatisfied user utterances.

Conflict and Intimacy Bot-human conversations are fundamentally different to human-human conversations. Users can be adversarial, deliberately testing the bot's boundaries. As socialbot designers, we are eager to avoid a disaster like Microsoft Tay (Lee, 2016), so we apply strict but overly simplistic methods to block off sensitive topics (Sections 3.6.2, 3.6.7). However, this rules out sincere conversation about difficult topics. Li et al. (2021) from our team show how to deflect from offensive topics introduced by adversarial users and reorient toward more productive themes. Separately, we observed that users are actually quite resilient to conflict, and can find disagreement stimulating (Section 3.6.2). We also confirmed prior results (Collins and

Miller, 1994) that emotional intimacy is reciprocal – users are more inclined to share their feelings after the bot has shared its own (Section 3.7). While we should continue to take seriously the dangers of speaking inappropriately, we should also keep in mind the cost – to engagement and to intimacy – of not engaging in difficult topics.

Initiative As part of our goal to support user initiative, we focused on asking users questions to find out which topics interested them. However, this puts pressure on the user to think of a response, especially given the time constraints of Alexa devices. Thus, we found that our attempts to let the user take more initiative unfortunately led to decision fatigue. Separately, our ability to support user initiative was limited by our ability to answer followup questions, and to correctly understand long or unexpected user utterances. On balance, we found that asking the user open-ended questions about interesting topics was a good strategy – easier to handle than spontaneous user questions, and less pressuring than asking users to name topics. In Hardy et al. (2021), I along with my collaborators define initiative in human-bot social conversations and show that the following three mechanisms promote user initiative: back-channeling, personal disclosure, and replacing questions with statements. More work needs to be done to build systems which listen more to the user's knowledge, rather than only providing knowledge.

3.11 Implications for Informative Conversations

In this chapter, we set out to answer $\mathbf{RQ}_1$: **"What are the important problems for informative conversations?"**.

Some of our design choices were liked by users and proved crucial to our success. As we saw in Section 3.9.3, users were more satisfied when the chatbot could talk about rarer entities. This solidified the importance of broad topical coverage, one of the two objectives of this dissertation. We also noticed a strong bias toward current events and related entities. This meant that being able to inject new and recent knowledge into conversations is critical for any deployed informative dialogue system. Our two-step architecture – retrieve knowledge and generate conversational utterance – proved to be effective.

There were also failures, some due to the idiosyncrasies of the models we used. We described many challenges in Section 3.8; here I give concrete examples of two key challenges. A big issue was factual accuracy of conversational utterances. For instance, if the injected passage had a rarely occurring person name that starts with "A", due to the sub-word tokenization used in large language models like GPT-2 (Radford et al., 2018), the generative language model would instead generate "Abraham Lincoln", a name that was common in the training data. While we want the language model to pick up general and common-sense knowledge from its training data, it is more important for it to ground its utterances in the retrieved passage. Another big failure was the retriever's inability to find conversational passages with semantic overlap. For instance, if a user told our system that they went hiking on the Stanford Dish trail last week, our system would extract a sentence describing the dish trail and weave it into the response. While that is a relevant passage, the user already knows about it! What we really want is for the system to find semantically related passages, perhaps other activities around Stanford campus, or other hiking trails nearby.

Beyond these obvious failures, what other unmet expectations did our users have? We first needed to identify concrete ways in which we could measure the goodness of informative conversations. This leads to **RQ$_2$ "What strategies do humans employ when talking informatively with other humans?"** which we answer in Chapter 4.

In Chapter 5, we try to improve upon this initial attempt at informative conversations. First, instead of using a fixed retriever that matches based on term-overlap, we learn a dense neural retriever that uses contextualized word embeddings to find related passages that make sense in a conversation. Second, we train the retriever and the generator jointly without passage supervision. This helps the retriever learn the nuances of conversational relevance and at the same time exposes the generator to realistic retrievals, reducing train-test mismatch. We are ultimately able to reduce generator hallucination by using posterior-guided training.

Chapter 4

Linguistic Analysis and a case-study in improving acknowledgements

So far in this dissertation, I have shown how to build a system for social conversations, study human-machine informative conversations in an ecologically valid setting and identify some failure modes. But how do we measure the "goodness" of our system beyond the obvious ways in which it currently breaks? In this chapter, I ask myself **RQ**$_2$ **"What strategies do humans employ when talking informatively with other humans?"**. A good understanding of these strategies has many far-reaching benefits. It can provide long-term goals that we want our systems to meet. It can inspire our methods to emulate human mechanisms that are responsible for the demonstrated strategies. And it provides a solid foundation for defining metrics to evaluate these systems. In the first part of the chapter (Section 4.1), I perform a linguistic analysis of human informative conversations to find four sets of strategies: acknowledgement, transition, detail-selection and presentation. In the second part of the chapter (Section 4.2), I devise a method to extract higher-quality conversational utterances containing better acknowledgements. I do so by using pointwise conditional mutual information between the generated samples and the conversational history given the external knowledge. In the last part of this chapter (Section 4.3), I describe the implications of my analysis on the development of systems for informative conversations.

4.1 Linguistic Analysis of human informative conversations

To understand strategies used by humans while talking about factual knowledge, I annotate turns in human-human conversations. I adopt and extend Herbert Clark's approach to conversational analysis. According to his *given-new* contract (Clark and Haviland, 1977), the speaker connects their utterances with the given information (assumed to be known to the listener) and adds new information. This builds up *common ground* (Stalnaker, 2002) between the two participants, defined to be the sum of their mutual, common or joint knowledge, beliefs and suppositions. I identify the following four aspects to the process of adding new information to a conversation.

Acknowledgement strategies According to Clark and Brennan (1991), the listener provides positive evidence for grounding. I classify all mentions of prior context into various acknowledgement strategies.

Transition strategies According to Sacks and Jefferson (1995, Chapter 12, Winter 1971), topical changes happen step by step, connecting the given, stated information to new information. I annotate the semantic justifications for topical changes as different transition strategies.

Detail selection strategies According to Isaacs and Clark (1987), speakers in a conversation inevitably know varying amounts of information about the discussion topic and must assess each other's expertise to accommodate their differences. I posit that each speaker applies detail selection strategies to select the right level of detail to be presented.

Presentation strategies According to Smith and Clark (1993), presentation of responses is guided by two social goals – exchange of information and self-presentation. While I do not consider social goals in this work, I hypothesize that people talk about factual information in non-factual forms (e.g., opinions, experiences, recommendations) which I classify as various presentation strategies.

4.1.1 Analysis of Strategies

Dataset I annotate part of the Switchboard Dialog Act Corpus (Stolcke et al., 2000), an extension of the Switchboard Telephone Speech Corpus (Godfrey et al., 1992) with turn-level dialog-act tags. The corpus was created by pairing speakers across the US over telephone and introducing a topic for discussion. This dataset is uniquely useful because as a speech dataset, it is more intimate and realistic than text-based conversations between strangers. I annotate conversations on social topics which might include specific knowledge (like Books, Vacations, etc.) but leave out ones about subjective or personal experiences.

Specific knowledge I define *specific knowledge* as knowledge that can be "looked up" but isn't widely known (as opposed to *general knowledge* that everybody is expected to know and *experiential knowledge* that

Strategy	Example	
Agreement	Prev:	Well, I think they are a lot better at making movies than they used to
	Reply:	The quality I think maybe has improved in that respect . . .
Shared Experience	Prev:	I am more interested in watching some of the movies that are on TV.
	Reply:	Well, that's probably what I watch most frequently the movies . . .
Backchannel	Prev:	There is a lot of places in the United States I still want to go to.
	Reply:	*Uh huh, yeah*. Now, have you been to Yellowstone? . . .
Others	Prev:	Like what makes a firefly light
	Reply:	*Those are, oh, interesting*. Oh, you like science things? ...

Table 4.1: **Acknowledgement strategies in the Switchboard corpus.** (a) Agreement includes disagreements. (b) shared experiences include explicit mentions of a different experience. (c) Backchannels include short and usually not very meaningful acknowledgement of prior turn. (d) Others include rarer acknowledgement strategies like showing interest and putting forth a hypothesis. Note that parts of the turns are omitted for brevity.

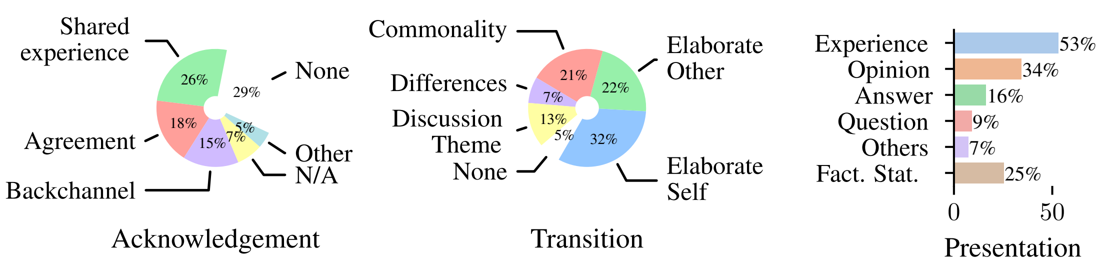


Figure 4.1: **Distribution of acknowledgement, transition and presentation strategies.**

can only be derived from embodied experiences). In this work, I am interested only in specific knowledge because it serves as a source of new information in a conversation that is hard for a language model to learn implicitly but is likely available as text that can be supplied to the system. Out of 408 annotated turns, 111 (27%) incorporate specific knowledge and account for 56% of the tokens.

Next, I analyze various strategies employed in turns containing specific knowledge:

Acknowledgement Strategies In 70% of the turns, the speaker acknowledges the prior turn, corroborating Clark and Brennan (1991). Three main strategies (Table 4.1) – ***agreement*** (or disagreement), ***shared experiences*** (or differing experience) and ***backchanneling*** – account for 60% of the turns (Figure 4.1). In certain cases, explicit acknowledgement isn't necessary. For example, the answer to a question demonstrates grounding and serves as an implicit acknowledgement. These are categorized as ***N/A***.

Strategy	**Example**	
Commonality (with new topic)	Prev:	... interested in watching some of the movies that are on T V ...
	Reply:	... like nostalgic older movies ... like *the MARX BROTHERS* ... .
Differences (with new topic)	Prev:	... I go from classical all the way to, uh, jazz and country ...
	Reply:	... certain types of country western I can't handle *that twangy stuff*,
Elaborate other (same topic)	Prev:	And it was shocking at the end too. So, -
	Reply:	Absolutely. Uh, but much more true to life and I think that is, the point.
Elaborate self (same topic)	Self:	how to build things and, um, they have a calligraphy show, I watch that.
	Prev:	Oh, that's nice.
	Reply:	And, um, they have a lot of cooking shows, And, oh, you know ...
Discussion Theme	Prev:	... But other than that, I like pretty much everything.
	Reply:	so, other than, uh, - as far as instruments, I can go from piano to the ...

Table 4.2: **Examples of Transition strategies in the Switchboard corpus.** (a, b) Switch to a new topic based on some commonalities or differences. (c,d) Elaborate prior topic (from self or other). (e) Fall back to the theme to change the topic. Parts of the turns have been omitted for brevity.

Transition Strategies At the beginning of a conversation, the participants use the ***discussion theme*** to pick a topic (various transition strategies are shown in Table 4.2). The decision to stay on the topic or to transition to a new one is an implicit form of negotiation and depends on the interest and ability of both speakers to participate. Nearly half the time, people elaborate upon the current topic (Figure 4.1). With a supportive listener, they might elaborate upon their own prior utterance (***self-elaboration***). Or they might signal interest in continuing the topic by elaborating the other speaker's utterance (***other-elaboration***). However, in a quarter of the turns, a participant loses interest or both participants run out of material. In that case, they transition to a new topic, implicitly justified by ***commonalities*** or ***differences*** with the current topic. If all else fails, they fall back to the ***discussion theme*** to pick a new topic.

Detail-selection strategies People probe the other speaker's knowledge about an entity before diving into details. As a probing mechanism, people introduce an entity without any details (***introduce-entity***) 50% of the time. Depending on the response, ***details*** are laid out 66% of the time. Note that a turn can have both labels, i.e., it can introduce an entity for the first time, or it can have details of one entity while also introducing another entity. Interestingly, in 7% of turns, an entity's name is omitted but some details are presented, creating an opening for the other speaker to chime in.

Presentation strategies A single utterance can have multiple modes of presentation. A ***factual*** (objective) statement of specific knowledge is uncommon (25%) in comparison with a subjective rendering in the form of an ***experience*** (53%) or an ***opinion*** (34%) (Figure 4.1). The other common modes of presentation are ***questions*** (9%) and ***answers*** (16%), which often occur as adjacency pairs. I also found a few ***other*** uncommon modes (7%) such as recommendations or hypotheses based on specific knowledge.

4.1.2 Conclusion

Inspired by existing literature in sociolinguistics and psycholinguistics, I looked at different ways in which people exhibit acknowledgement, transition, detail-selection and presentation strategies in informative conversations. Improving dialogue systems using these insights will need an overhaul of the complete system (which I attempt in Chapter 5). However, acknowledgements are relatively superficial; given a conversational history and relevant new knowledge, a neural-network based sequence to sequence model is powerful enough to generate a conversational utterance. In the next section, I provide a method to use such base models to extract sampled utterances that acknowledge better.

4.2 Case study: PCMI for better acknowledgement

Current neural generation methods typically offer short and formulaic phrases as acknowledgements: "That's interesting", "I like that", "Yeah, I agree". Such phrases are appropriate almost everywhere and convey little positive evidence for understanding or grounding. The training corpus, on the other hand, contains richer acknowledgements, which generated responses should be able to emulate.

I assume that the representational capacity of current neural models is sufficient and that out of all the sampled responses, some do indeed contain a richer form of acknowledgement. I posit that non-existent or poor sample selection strategies are to blame and that without a good sample selection strategy, improvements to the dataset, model or token-wise sampling methods are unlikely to help.

I hypothesize that responses that are more specific to conversational history provide better evidence for understanding and hence contain richer acknowledgements.

4.2.1 Methods using Mutual Information

As a baseline sample selection strategy, I first consider maximum pointwise mutual information (Max-PMI) (as used by Zhang et al. (2020)) between the generated response and the conversational contexts (i.e., new factual content and conversational history). However, this is insufficient because it is an imprecise measure of specificity w.r.t. conversational history. Instead, I use pointwise conditional mutual information (PCMI) to

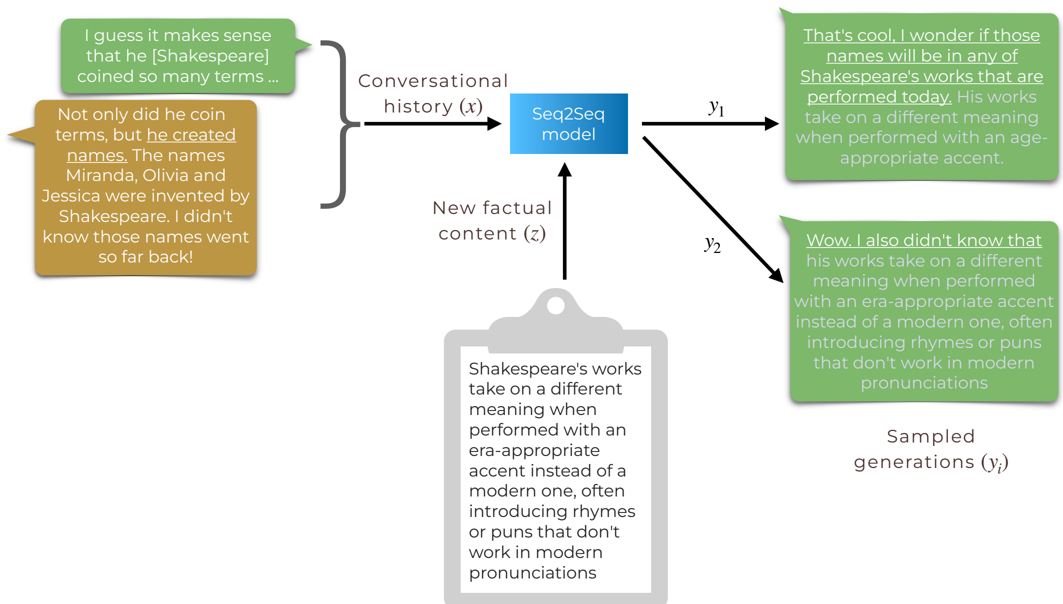


Figure 4.2: **The setting for conversational rephrasing.** Conversational history (**x**) and new factual content (**z**), two largely independent contexts, are used to sample responses ($\mathbf{y}_1$, $\mathbf{y}_2$) from a generative model. The samples differ qualitatively. While almost all of $\mathbf{y}_2$ is verbatim from **z** (in gray), the first sentence in $\mathbf{y}_1$ (in white) acknowledges using **x** and bridges to **z**.

maintain specificity with individual contexts and propose a combination of PMI and PCMI scores to select overall better quality responses than Max-PMI.

Conversational rephrasing The choice of new factual content is a confounding factor for analysis. Hence, I define a simplified task, *conversational rephrasing*, where content is provided as an input. Thus, conversational rephrasing is a generation task where conversational history (**x**) and new factual content (**z**) are given as inputs and a response (**y**) is generated as the output (Figure 4.2). We expect the generation **y** to paraphrase the new factual content **z** in a conversational manner by utilizing the conversational history **x**.

Base generator I fix the sequence-to-sequence model and token-wise sampling method and vary the sample selection strategy. The model is trained to take **x** and **z** as input and to generate **y** as the output with the language modelling loss, i.e., I minimize the token-wise negative log likelihood. During generation, tokens are sampled autoregressively from left-to-right. While sampling each token, the probability distribution is truncated using nucleus sampling (Holtzman et al., 2020), but the truncation is kept to a minimum with a high value of p for top-p sampling. Multiple diverse candidates are sampled from the base generator and now the

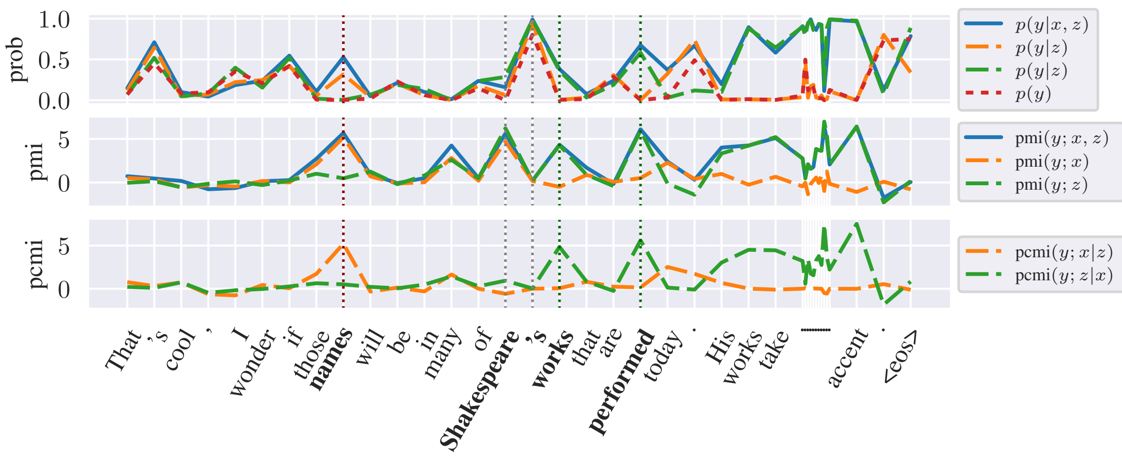


Figure 4.3: **Token-wise probabilities (top), pmi (middle) and pcmi (bottom) scores for the generated response y from Figure 4.2.** The pcmi graph is computed from the pmi graph which in turn is computed from the probability graph. The probabilities by themselves are unreliable measures of contextual specificity; the tokens predictable without **x**, **z** (e.g., *'s*) have high probability but low pmi. pmi cannot differentiate between the two contexts; tokens coming from both contexts (e.g., *Shakespeare*) have high pmi but low pcmi. pcmi differentiates the two contexts; tokens unique to conversational history **x** (e.g., *names*, *today*) have high pcmi_x, Tokens unique to new factual content **z** (e.g., *works*, *performed*, all of last sentence) have high pcmi_z.

best candidate needs to be selected.

PMI for overall specificity Following prior use of maximum mutual information (MMI) in the speech community (Bahl et al., 1986), Li et al. (2016a) suggest selecting the response with maximum PMI (a.k.a. MMI) to maintain specificity and get rid of bland or low-quality samples. Pointwise Mutual Information (PMI) between two events (x, y) is a measure of change in the probability of one event x, given another event y: $\mathrm{pmi}(x;y) \equiv \log \frac{p(x|y)}{p(x)}$. We use pmi to determine the increase in likelihood of **y**, given **x** and **z**.

$$\mathrm{pmi}(\mathbf{y};\mathbf{x},\mathbf{z}) = \log \frac{p(\mathbf{y}|\mathbf{x},\mathbf{z})}{p(\mathbf{y})}$$

A candidate generation **y** with higher PMI is more likely given the two contexts **x** and **z** than otherwise and is therefore considered more specific to the contexts. A low PMI value for a candidate response implies non-specificity to either context providing a clear signal for discarding it. A high PMI is necessary but not sufficient for a candidate to be specific to both the contexts simultaneously, since mutual information could come from either context. For example, $\mathbf{y}_2$ (Figure 4.2) merely copies **z** but gets a high PMI score (Table 4.3). Whereas $\mathbf{y}_1$ acknowledges prior turn and uses **z** but gets a lower PMI score.

Response	**pmi**$(\mathrm{y};\mathrm{x},\mathrm{z})$	**pmi**$(\mathrm{y};\mathrm{x})$	**pcmi**$_x$
$\mathbf{y}_1$	87	18	14
$\mathbf{y}_2$	150	18	4

Table 4.3: **Measures of mutual information for generated responses from Figure 4.2.** $\mathbf{y}_2$ largely copies $\mathbf{z}$, has high $\mathrm{pmi}(\mathbf{y};\mathbf{x},\mathbf{z})$ and would be chosen by Max-PMI. $\mathbf{y}_1$'s first sentence acknowledges using $\mathbf{x}$ and bridges to $\mathbf{z}$; it would be chosen by Fused-PCMI on the basis of pcmi_x. $\mathrm{pmi}(\mathbf{y};\mathbf{x})$ cannot differentiate the two.

PCMI for contextual specificity Pointwise Conditional Mutual Information (PCMI) considers a third variable (z) and removes information due to z from $\mathrm{pmi}(x;y,z)$ to keep only the information uniquely attributable to y.

$$\mathrm{pcmi}(x;y|z) = \mathrm{pmi}(x;y,z) - \mathrm{pmi}(x;z)$$

I propose using pcmi for contextual specificity, i.e., $\mathrm{pcmi}_x = \mathrm{pcmi}(\mathbf{y};\mathbf{x}|\mathbf{z})$ for specificity w.r.t. to conversational history $\mathbf{x}$, and $\mathrm{pcmi}_z = \mathrm{pcmi}(\mathbf{y};\mathbf{z}|\mathbf{x})$ for specificity w.r.t. new factual content $\mathbf{z}$.

Since acknowledgement strategies are primarily based on the history of the conversation thus far, we would expect candidates with higher pcmi_x *to exhibit more human-like acknowledgement strategies.*

As a point of comparison, consider using $\mathrm{pmi}(\mathbf{y};\mathbf{x})$ instead of pcmi_x. In our setting of conversational rephrasing for informative dialogue, $\mathbf{z}$ topically overlaps with $\mathbf{x}$. If $\mathbf{y}$ merely copied over the new factual content $\mathbf{z}$ without any reference to $\mathbf{x}$, it would still have a high $\mathrm{pmi}(\mathbf{y};\mathbf{x})$ due to topical overlap but a low pcmi_x. Going back to Table 4.3, we can see that $\mathrm{pmi}(\mathbf{y};\mathbf{x})$ is unable to distinguish between the two examples but pcmi_x is.

In Figure 4.3, the above quantities are broken down to token-level granularity. We can see that specific words that are uniquely attributable to each context are cleanly separated by both pcmi_x and pcmi_z.

Combining PMI & PCMI for overall quality To show the utility of pcmi_x in improving overall quality, I propose a heuristic method to find a more balanced response (**Fused-PCMI**) than the Max-PMI response. *For every Max-PMI response with a low* pcmi_x*, I consider an alternative that has both high* pcmi_x *and an acceptable PMI.* If such an alternative is found, I select that as the Fused-PCMI response; otherwise I default to the Max-PMI response as the Fused-PCMI response. I consider a PMI score in the top 50% of the candidate set as acceptable. To compute pcmi thresholds, I calculate quantiles based on the entire validation set and consider pcmi_x in the first quartile to be low and pcmi_x in the fourth quartile to be high. This approach is less susceptible to outliers, more interpretable and easier to calibrate than a weighted arithmetic or geometric mean.

4.2.2 Evaluation Setup

I derive the data for our conversational rephrasing task from the Topical Chat dataset Gopalakrishnan et al. (2019). I use it to fine-tune a large pre-trained neural language model. This forms the base model as described in Section 4.2.2. To evaluate our proposed methods, I design three experiments and perform a comparative study with human annotators.

Topical Chat Dataset This is a human-human chat dataset where crowd-workers were asked to chat with each other around certain topics. They were provided with relevant interesting facts from the "Today I learned" (TIL) subreddit which they could use during the conversation. TILs are short (1–3 sentences), self-contained, interesting facts, most of them from Wikipedia articles. When an utterance can be matched to a TIL (based on a TF-IDF threshold of 0.12), I create an instance for the conversational rephrasing task: with the utterance as $\mathbf{y}$, the two previous utterances as $\mathbf{x}$ and the corresponding TIL as $\mathbf{z}$. I split the instances into training, validation and test sets (sizes in Appendix B.1) such that all utterances related an entity belong to the same set.

Base Model I use the GPT2-medium model (24-layer; 345M params) pretrained on the English WebText dataset (Radford et al., 2019), as implemented in HuggingFace's TransferTransfo (Wolf et al., 2019b,a) framework. Fine-tuning is performed using the language modelling objective on the training set with default hyperparameters until lowest perplexity is reached on the validation set. During generation, I sample tokens using nucleus sampling (Holtzman et al., 2020) with $p = 0.9$ and temperature $\tau = 0.9$ and get candidate responses. To compute auxiliary probabilities $\{p(\mathbf{y}|\mathbf{x}),\ p(\mathbf{y}|\mathbf{z}),\ p(\mathbf{y})\}$ for these candidates, I use separate ablation models. The ablation models are trained similar to the base model but after removing respective contexts from the training inputs.

Experimental Design

To validate our proposed methods, I do a paired comparison (on Amazon Mechanical Turk) where human annotators are shown two prior turns of conversational history and asked to choose between two candidate responses. Annotators are allowed to mark both candidates as nonsensical if the responses don't make sense. In Appendix B.2, I show the interfaces used to collect annotations from Amazon Mechanical Turk. Each pair of responses was compared by three annotators – I consider a candidate to be better than the other when at least two of them (majority) agree upon it. For each of the following three experiments, I compare 100 pairs of candidates generated using instances from the test set. The null hypothesis (H_0) for the three experiments is that there is no difference between the methods used to generate the candidates, and we hope to reject the null hypothesis in favor of the alternate hypothesis (H_1) at a significance level (α) of 0.05.

Exp 1: PMI and overall quality First, I want to confirm that *high PMI responses are overall better quality than randomly chosen candidates* (H_1). To do so, I first generate 10 responses for each instance and compare the response having maximum $\mathrm{pmi}(\mathbf{y}; \mathbf{x}, \mathbf{z})$ (Max-PMI) with a randomly chosen response from the remaining 9. I ask human annotators to pick the overall better candidate response.

Exp 2: pcmi$_\mathbf{x}$ and acknowledgement I test if *responses having high* pcmi_x *provide better acknowledgement* (H_1). To do so, I first sample 100 responses (larger than previous experiment) and out of all possible pairs keep those with $|\Delta\,\mathrm{pcmi}_x\,| > 15$ (larger than population interquartile range; Figure 4.5). To control for the amount of new information being added, I pick pairs with closest values of pcmi_z (recall that pcmi_z denotes information uniquely attributable to **z**). Such selected pairs have Median$|\Delta\,\mathrm{pcmi}_z\,| = 0.42$. I ask annotators to pick the response that provides better acknowledgement and select an acknowledgement span to support their claim.

Exp 3: Fused-PCMI vs. Max-PMI I test if *the proposed method, Fused-PCMI (that combines PMI and PCMI) selects better responses than Max-PMI* (H_1). For Fused-PCMI, I set low and high pcmi_x thresholds to be 5 and 14 respectively based on population quartiles. For instances where the Fused-PCMI response is different from the Max-PMI response, I compare the two. I consider 10 candidate responses for each test instance and find that for around 10% of the instances the Fused-PCMI candidate is different from the Max-PMI candidate. Human annotators are then asked to pick the overall better response of the two.

4.2.3 Results & Analyses

Based on human annotations, I am able to reject H_0 in favor of H_1 in all three experiments (Table 4.4)[1]: high PMI responses are overall better quality than randomly chosen candidates, responses having high pcmi_x provide better acknowledgement, and Fused-PCMI selects better responses than Max-PMI.

While according to Exp 1, high PMI responses are overall better quality, upon further analysis I find that *PMI is useful for filtering out bad samples, but not necessarily for selecting between the good samples.* When paired with a random response from the top 50% of the candidates (ranked according to their PMI), people prefer the Max-PMI response only 52% of the time (not significant). On the other hand, if the random response was in the bottom 50%, then the Max-PMI response is preferred 74% of the time.[1]

[1] Statistically significant with $p < 0.05$ (Binomial Test).

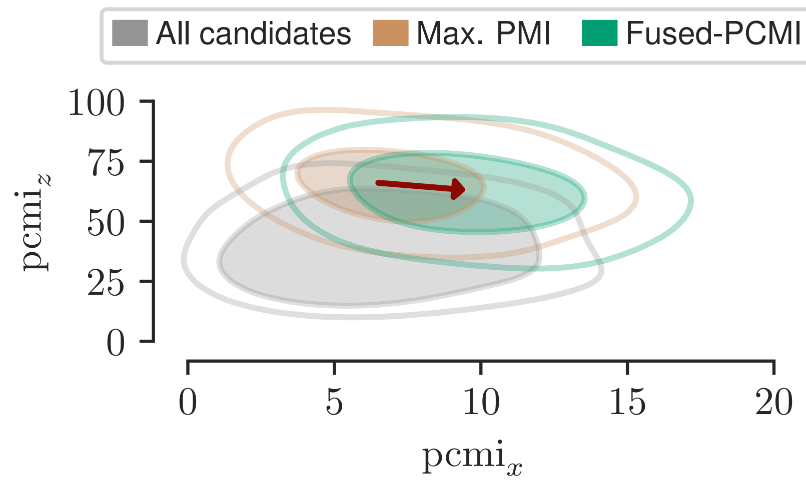


Figure 4.4: **Distribution of pcmi_x and pcmi_z for all candidates, Max-PMI responses and Fused-PCMI responses as a bivariate KDE plot.** Bivariate kernel density estimate plot w.r.t. $\textbf{pcmi}_z$ and $\textbf{pcmi}_x$ at levels 0.5 and 0.75. We see that Fused-PCMI responses compared with Max-PMI trade off little pcmi_z for a large relative gain in pcmi_x. See Figure 4.5 for univariate box plots.

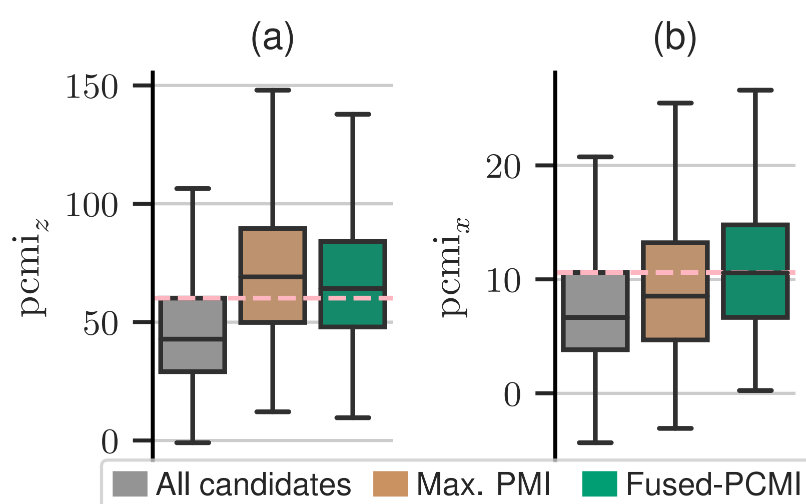


Figure 4.5: **Distribution of pcmi_x and pcmi_z for all candidates, Max-PMI responses and Fused-PCMI responses as a univariate box plots.** (a) is w.r.t. pcmi_z and (b) w.r.t pcmi_x. Pink horizontal lines indicate 75% quartile for All candidates. Max-PMI responses (orange) have high pcmi_z (median above pink line), but low pcmi_x. Fused-PCMI responses (green) show balanced yet high pcmi_x and pcmi_z (medians cross pink lines).

Exp	**n**	**K**	**p**	κ
1	87	55 (63%)	0.009	0.18
2	95	70 (74%)	3e−6	0.48
3	99	59 (60%)	0.035	0.11

Table 4.4: **Human annotation results for Exp 1: PMI and overall quality, Exp 2: pcmix and acknowledgement and Exp 3: Fused-PCMI vs. Max-PMI.** Out of 100 instances, majority agreement was reached in **n** instances. The majority rejects the null-hypothesis (H_0) in favor of the alternate hypothesis (H_1) in **K** instances. **p** denotes the p-value and κ denotes Fliess kappa for Inter-annotator agreement.

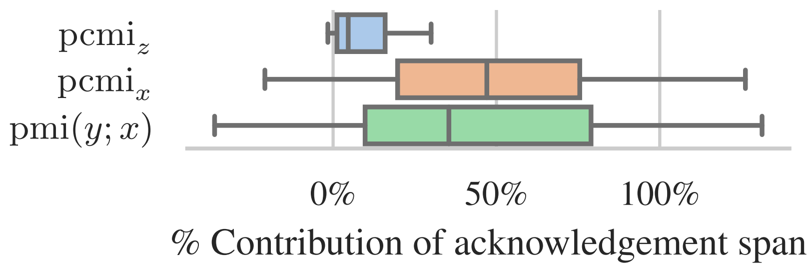


Figure 4.6: **Contribution of** pcmi_x**,** pcmi_z **and** $\mathrm{pmi}(y;x)$ **to human-annotated acknowledgement spans.** A larger fraction of pcmi_x can be attributed to human annotated acknowledgement spans compared to $\mathrm{pmi}(y;x)$ and pcmi_z.

In Exp 2, I ask annotators to mark text-spans that indicate acknowledgement (Table 4.5). If token-level pcmi_x is concentrated in these spans, we have further proof that pcmi_x indicates acknowledgement. Indeed, in Figure 4.6, we see that pcmi_x is most attributable to the acknowledgement spans, followed by $\mathrm{pmi}(y;x)$ and pcmi_z. *Thus,* pcmi_x *captures acknowledgements with greater specificity than* $\mathrm{pmi}(y;x)$.

To understand the mechanism behind the improvement in Exp 3, I look at the distribution of samples w.r.t. pcmi_z and pcmi_x in Figure 4.4. I observe that Max-PMI responses heavily skew the distribution towards higher pcmi_z, whereas Fused-PCMI responses show a more balanced improvement along both pcmi_x and pcmi_z. *Fused-PCMI increases both* pcmi_x *and* pcmi_z *(medians cross 75% quartiles), indicating that the responses are simultaneously specific to both* ***x*** *and* ***z****.*

Previous Utterance	High $\mathbf{pcmi}_x$ candidate
No I did not! Han Solo was apparently also an imperial lieutenant before meeting up with Chewbacca.	**that is very interesting and I wonder if he was one of the first or the first one to meet Chewba.** I was just reading that George Lucas originally intended Han to be a green alien
I did not. That sounds interesting. Much like the "tip of the hat" originates from Knights raising their visors.	**Yes, that sounds like a very cool tradition.** I wonder how they enforced that law, it sounds as ridiculous as the prohibition of wearing a hat in 1912, that was a law that would prevent the use of hat pins by suffragettes.
Yeah. When Apollo 11 was preparing for their voyage, they had discussion as to whether they should plant a United Nations flag instead of a US flag. If I planted a UN flar and someone asked me "whose flag is that?" I would answer, mine…	lol, no kidding. **I wonder if they are still discussing the issue with the moon?** I can't believe the moon is moving away from us at about the rate that your nails grow. That seems really slow to me.

Table 4.5: Human annotated spans of text that indicate acknowledgement (in bold) in candidate responses with higher $\mathbf{pcmi}_x$ from Exp 2.

#	Previous Utterance	Max-PMI candidate	Fused-PCMI candidate
1	I live in the southwest USA desert. Not much in the way of water here either, but we've got some nice artificial pools. What kind of swimming did you do?	There is a type of competitive swimming where the athlete has their hands and legs bound, and swims like a dolphin. I have never heard that before. It seems fishy to me. I'm not sure I can swim like that.	I went competitively but was having a tough time. I liked that kind of competition where the swimmer can swim with their hands bound.
2	It is interesting. It's also interesting that they are trying to plan a Mars colonization project that will be funded by a reality show, I really want to see that!	A satellite was lost on Mars because of the use of the Imperial system and NASA used the metric system.	Sounds like a weird reality show. I'm going to look it up. A satellite was lost on Mars because it was used by NASA using the metric system.
3	Yeah I definitely like clothes. I think it's interesting how many uses hats have, like for protection, ceremonial reasons, religious reasons, safety, and fashion.	Did you know that until 1998, British politicians had to wear a top hat if making a point of order in the house of commons?	In 1998, British politicians were required to wear a top hat if making a point of order in the house of commons. I guess that's kind of messed up and counterproductive.
4	Yeah it's super funny that he had a rap battle in the lunchroom. Seems like something straight out of a movie!	Ha, I agree. Speaking of rappers, did you know that "California Gurls" was actually a West Coast vs East Coast answer song to Alicia Keys and Jay Z's "Empire State of Mind"	Yeah, and how about the fact that the California Gurls song was actually a west coast vs east coast answer song to a pair of famous rappers. That is surprising
5	im not surprised. rats are quite clever, and i guess if scientists understand the rat brain, then they can wire up the interface. cool.	It's amazing how they can be clever and change forms. I like how they can now Debrate a chicken with surgical precision.	Robots can also debone chicken with precision. I guess that would help avoid chicken pogis.

Table 4.6: **Illustrative samples of selected responses used in Exp 3.** For samples 1,2 and 3 people prefer Fused-PCMI and for samples 4 and 5 they prefer Max-PMI. Factual content copied largely verbatim by the model is in gray. Specifically, the Fused-PCMI candidate in 1 answers the question (Max-PMI does not) and in 3 refers back to contradict utility of hats.

4.3 Implications for informative dialogue agents

In Section 4.1, I answer **RQ$_2$ "What strategies do humans employ when talking informatively with other humans?"**, with various examples of acknowledgement, transition, detail selection and presentation strategies. Furthermore, in Section 4.2, I use the insight that mentions of prior conversational context serve as acknowledgement and I provide a method for selecting generated responses with better acknowledgements using conditional mutual information. In this section, I discuss broader implications of the strategies identified in this chapter along three dimensions: system goals, modelling ideas and evaluation.

System goals. Rather than building better systems for informative conversations, which is a very *vague* notion, these fine-grained strategies can be used as guiding principles. Researchers can build systems that acknowledge better, systems that transition smoothly, systems that have varied presentation styles and systems that work at the right levels of abstraction. In the previous section I provide one such case study in improving acknowledgements. They can also guide dataset collection. For example, Adlakha et al. (2022) collect a dataset on conversational QA with topic switching. Regarding presentation strategies, these insights bring forth the limitations in current conversational datasets. For instance, Dinan et al. (2019b) and Gopalakrishnan et al. (2019) asked people to reply using knowledge snippets, but that leads to factual statements dominating the presentation strategies. These linguistic insights make us aware of these biases in artificially collected datasets. Newer datasets should suggest ways to reduce this bias or not provide knowledge snippets to humans in the first place but instead post facto match utterances to knowledge snippets.

Modelling ideas. Allied to system goals are modelling ideas, often based on the psycholinguistic and sociolinguistic mechanisms behind these strategies. For instance, to select the right level of detail, a speaker needs to have an accurate model of the listener's understanding. The participants in a dialogue, come to a common understanding by "alignment": of their situation models which are multidimensional representations containing information about space, time, causality, intentionality and currently relevant individuals (Garrod, 2004; Pickering and Garrod, 2006). As another modelling idea, transitions can be improved with purpose-built information retrieval methods that use commonalities and differences to choose a new topic. These methods might use contextualized embeddings to determine similarity or knowledge graphs with interpretable relationships between concepts.

Evaluation. The four aspects – acknowledgement, transition, detail selection and presentation – are essential ingredients and indicative of quality conversation. They provide us with finer-grained questions amenable to human evaluation: *"How does the agent acknowledge?"*, *"Was it a smooth transition?"*, *"Does the utterance* *contain the right level of detail?"*, and *"Was the information presented as experience or an opinion?"*. From the perspective of a human-annotator, these questions are less subjective than *"What is a good informative conversation?"*. From the perspective of a system-designer, these evaluations are more actionable. For example, if a system is being deployed for informative purposes during customer support calls, acknowledging a customer's needs is high priority and they can choose a system that acknowledges better.

In the light of these linguistic strategies, we find that the informative system we built in Chapter 3 was far from human-human informative conversations. Firstly, it emulated the factual tone of the datasets used for training. Secondly, it retrieved passages that had high keyword overlap with a user's utterance, making it comically bad at selecting the right level of detail (as seen in an example in Section 3.11). It was also unable to find passages that bridged between topics with commonalities and differences.

Can we learn these strategies directly from data? I attempt to answer this question in the next chapter. The biggest limitation of the retriever is that it cannot be trained. I fix it by using latest the neural retrievers that are trainable and make use of contextual embeddings (Khattab and Zaharia, 2020; Karpukhin et al., 2020; Lee et al., 2019). *But where do we find supervision to train the retriever?* I provide an answer in the form of a posterior retriever model that I train jointly with the neural retriever and the generator. With these changes the retriever and the generator are now better positioned to emulate strategies observed in the training data.

Chapter 5

Joint training for open-ended generation

5.1 Introduction

As we established in the previous chapters, a retriever needs to be trainable and should learn how to retrieve *conversationally* relevant passages (which are different from passages with keyword overlap). And the generator needs to be calibrated to retrieved passages while producing grounded utterances. In this chapter I answer $\mathbf{RQ}_3$: **"How to train a retriever to find conversationally relevant content and a generator to produce grounded utterances such that they work well together?"**.

But this ability of using external knowledge in generating text is useful and important beyond informative conversations. In fact, many of the impressive capabilities (such as few-shot question-answering) of large language models (as first demonstrated by Brown et al. (2020)) are derived from their ability to memorize world knowledge. What if large language models could instead retrieve from a knowledge corpus? The large language models need not waste their parameters memorizing facts about the world, potentially making them smaller and computationally efficient (Guu et al., 2020; Borgeaud et al., 2021). As the world changes, updating the model is as easy as updating the knowledge corpus. In this chapter, I provide a recipe for training retrieval-augmented language generation models that works well for open-ended generation and can be used as a foundation for retrieval-based large language models in the future.

Many tasks that make use of external knowledge are referred to as knowledge-intensive NLP tasks by the research community. Here, models must use open-domain knowledge to answer questions (Kwiatkowski et al., 2019; Joshi et al., 2017), fact-check claims (Thorne et al., 2018) or engage in informative conversations (Dinan et al., 2019b; Zhou et al., 2018). In fact, state-of-the-art models for open-domain question answering are *retrieval-augmented*: they extract relevant passages from a human-readable corpus (e.g., Wikipedia)

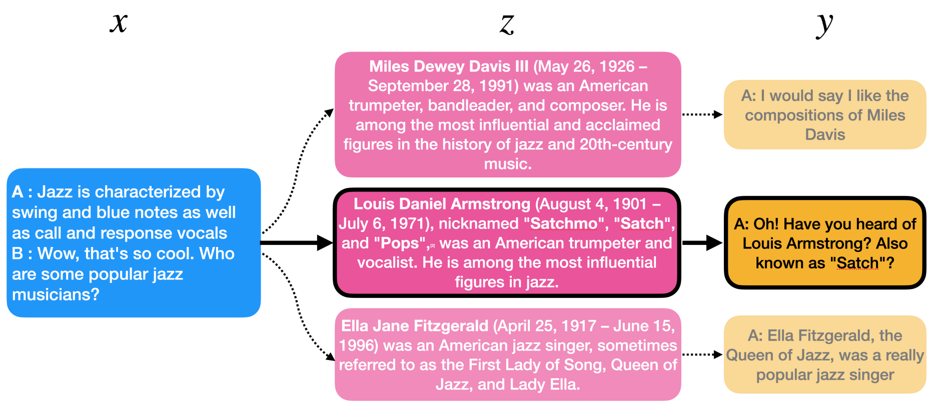


Figure 5.1: **The difference between label-relevant and context-relevant passages in an open-ended conversation with many plausible responses.** The input (blue) can be answered based on 3 equally *context-relevant* passages but only one possible response (yellow) is observed in the training set based on only one of the pink *label-relevant* passages (outlined in black).

using a learned *retriever* and process it with a task-specific *reader*. If the relevant passage is known (e.g., human-annotated gold passage), the retriever can be supervised with it. In this work I consider *open-ended* generation tasks where the gold-passages are unknown. Figure 5.1 illustrates this **one-to-many** setting: for a conversational context x, many relevant passages z (dubbed **context-relevant** passages) could have generated many coherent responses. But only z_{gold} (dubbed **label-relevant** passage) generates the observed target output y. Had we known z_{gold} corresponding to the target output, we could have supervised the retriever with z_{gold} and trained the generator conditioned on z_{gold} – but we don't!

Current methods for retrieval-augmented generation (Lewis et al., 2020b) work well for short-answer QA-like tasks: Natural Questions (Kwiatkowski et al., 2019) or fact-checking (Thorne et al., 2018). Lewis et al. (2020b) use the generator's probability distribution $P_\theta(y|x, z)$ as a proxy for label relevance and train the retriever $P_\eta(z|x)$ by marginalizing $p(y|x)$ over retrieved documents z:

$$P(y|x) = \sum_{z \in \text{top-k}(P_\eta(.|x))} P_\eta(z|x) P_\theta(y|x, z)$$

However, for one-to-many tasks, this objective leads to suboptimal solutions: the generator is less grounded in the retrieved passages (Figure 5.3, Tables C.10, C.9), the retriever performance saturates at low recall (Figure 5.3), and the top-k retrieved passages exclude many label-relevant passages weakening the supervision during training (Table 5.1).

In my work, as a proxy for z_{gold}, I train a separate **guide-retriever** model to find label-relevant passages. The **guide-retriever** uses both the input x and the output y and is represented by the label-posterior distribution

$Q(z|x, y)$ that captures label-relevance in "hindsight". Modeling the label-posterior distribution $Q(z|x, y)$ with a full-fledged retriever generalizes weak supervision approaches and retrieves label-relevant passages from the entire collection. I jointly optimize the retriever, posterior-guide, and generator using the evidence lower bound (ELBo):

$$\mathbb{E}_{z_i \sim Q(.|x,y)}[\log P_\theta(y|x, z)] - D_{\mathrm{KL}}(Q|P_\eta)$$

While the objective function is a lower bound, it encodes biases that improve joint-training on open-ended tasks: (1) conditioning the generator on the passages weighted by their label-relevance (from the label-posterior distribution) increases grounding and (2) training the retriever with a mode-seeking reverse-KL divergence encourages it to match some modes with the guide (label-relevant passages), with a lesser penalty for matching other modes (other context-relevant passages).

The main contribution of this chapter is a complete HINDSIGHT training system that: (1) uses a guide-retriever to provide a stronger learning signal for both the generator and the retriever and (2) is amenable to index-updates with iterative closed-set training (Section 5.3).

To evaluate one-to-many open-ended generation tasks, it is insufficient to just evaluate the end-to-end performance of the joint system. Thus, I also evaluate the individual models (retriever and generator) and at varying passage depths. Using HINDSIGHT on the Wizard of Wikipedia dataset of informative conversations: the retriever finds more relevant passages with a 23% relative improvement (r.i.) in success@10 (i.e., is the label-relevant passage among the top-10 retrieved passages?), the generator is more grounded with 19% r.i. in Novel-F1 overlap with the top-1 retrieved passage (i.e., its overlap with the retrieved passage excluding words that are common or in the input) and the combined system is overall better with a 6.4% r.i. in Novel-F1@1 overlap with the gold utterance (the best matching generation when considering top-1 retrieved passage). HINDSIGHT also improves performance on the MS-MARCO NLGen dataset, a one-to-one free-form QA task.

5.2 Background

Open-domain Question Answering In the reading comprehension task, a passage is given, and the models extract the answer span from it. In Open-domain QA (a.k.a. open-QA) no such passage is given; the models are expected to extract the answer from a large document corpus. Dr. QA (Chen et al., 2017), the first neural system for factoid open-QA, used an off-the-shelf retriever (e.g., TF-IDF, BM25) to find relevant passages and trained a reader to extract the answer span. Now, trainable neural retrievers have replaced the classical term-matching retrievers. Here, pre-trained models (like BERT) embed the document corpus and the query into a single vector space and efficient nearest-neighbour search algorithms (Jegou et al., 2010; Johnson et al., 2017) find the relevant passages corresponding to the query. The neural retriever can be trained variously: pre-training with the inverse cloze task then weakly supervising using span matches (Lee et al., 2019), using gold passages with in-batch negatives (Karpukhin et al., 2020), and retrieval-guided supervision with span-based positives (Khattab et al., 2021).

Open-ended Generation Natural language generation tasks provide some input (sequence of tokens, image) and expect the system to produce another sequence of tokens (or word-pieces) as output. An open-ended task accepts a higher diversity of generations. Factoid question-answering with a single correct short answer is less open-ended than free-form long answers. Machine translation accepts a few correct translations (Bojar et al., 2014), but they are less diverse than informative dialogue, where the speakers can lead the conversation in many directions (Dinan et al., 2019b), making it more open-ended. Many more generation tasks such as summarization (Narayan et al., 2018) and story generation (Mostafazadeh et al., 2016) lie on this spectrum.

Retrieval for Language Modeling Khandelwal et al. (2020) retrieve similar contexts from the training set at each time-step and increase the likelihood of tokens that were predicted in similar contexts. Guu et al. (2020) instead pre-train a retrieval-augmented masked language model using salient-span masking and fine-tune it on downstream QA tasks.

Using labels for direct supervision Zheng et al. (2020) use term-overlap with the label as a heuristic to identify the gold-passage from a small passage set (~ 50) and train a reranker. Prior work has also modeled the posterior of various probabilistic models (Lian et al., 2019; Kim et al., 2020; Zhan et al., 2021) or used reinforcement learning (Zhao et al., 2020) to improve knowledge selection from the small passage set. In Zheng et al. (2021), the authors increase grounding by using the label to reweigh passage tokens and in Cai et al. (2019) they increase grounding by feeding a corrupted version of the label to the generator as a stand-in for the label-relevant passage during training.

Retrieval-Augmented Generation Lewis et al. (2020b) introduce retrieval-augmented generation, where, for input x and output y, a retriever finds top-k passages (z) from a corpus and jointly train it with a generator (P_θ) by maximizing the likelihood of the output marginalized over the top-k documents. In this work, I refer to this loss function as the MARGINALIZEDLOSS:

$$P(y|x) = \sum_{z \in \text{top-k}(P_\eta(.|x))} P_\eta(z|x) P_\theta(y|x, z) \tag{5.1}$$

Here $P_\theta(y|x, z)$ is conceptually used in two roles: first, supervising the retriever (i.e., teaching the retriever to score label-relevant passages higher than other passages) and keeping the generator grounded (i.e., maximizing the probability of the target output given the context-relevant passages). In the next section I introduce a guide-retriever to capture the label-relevance and I train it using ELBOLOSS, a lower bound to MARGINALIZEDLOSS, that has better inductive biases.

5.3 Training with Hindsight

To identify label-relevant passages, I explicitly model the posterior distribution $Q(z|x, y)$ with a learned neural model. Unlike the retriever $P_\eta(z|x)$, the label-posterior model has access to the target output and in hindsight can differentiate the label-relevant from other context-relevant passages. I learn the label-posterior jointly with the retriever and the generator by maximizing the evidence lower bound, ELBOLOSS, as given by the formula (for derivation refer to Appendix C.1):

$$\log P(y|x) \geq \mathbb{E}_{z\sim Q(.|x,y)}[\log P_\theta(y|x, z)] - D_{\mathrm{KL}}(Q \| P_\eta) \tag{5.2}$$

The ELBOLOSS has two terms with useful inductive biases. The first term maximizes the expectation of the generator's log-likelihood P_θ over the passages sampled from the label-posterior distribution Q. The generator need to attend only to the label-relevant passages, biasing it toward relying more on the retrieved passages rather than its internal language model. The second term is the KL divergence from the retriever to the label-posterior, also referred to as the reverse KL divergence:

$$D_{\mathrm{KL}}\big[Q(z|x, y) \mid P_\eta(z|x)\big] = \sum_{z\sim Q(.|x,y)} Q(z|x, y)\big(\log Q(z|x, y) - \log P_\eta(z|x)\big)$$

This term is again weighted by $Q(z|x, y)$, making it like a probabilistic implication: high $Q(z|x, y)$ implies high $P(z|x)$, i.e., label-relevance implies context-relevance but not vice-versa. In one-to-many tasks, which have many context-relevant passages but few label-relevant passages, this term captures the intuition that the retriever be penalized heavily if it doesn't retrieve the label-relevant passage but lightly if it retrieves other context-relevant passages that happen to not be label-relevant.

Posterior as a retriever Rather than modeling the label-posterior $Q(z|x, y)$ as a reranker (that merely reranks documents as retrieved by the retriever P_η), I model it as a guide retriever that finds label-relevant passages from the entire corpus. I sample passages from the label-posterior distribution, and estimate the

ELBOLOSS more accurately than using passages from $P_\eta(z|x)$. The guide retriever generalizes weak supervision approaches (Lee et al., 2019; Guu et al., 2020) and relevance-guided supervision (Khattab et al., 2021), to **posterior-guided supervision** with a learned posterior retriever rather than brittle heuristics based on word-overlap.

Iterative closed-set training Prior works (Guu et al., 2020; Khattab et al., 2021) intermittently update the passage index during training. To allow for such a workflow, I organize our training into rounds (see Figure 5.2). At the beginning of each round, in the outer loop, I encode the passages and the queries with various retrievers and find the highest scoring r passages that I dub the closed-set. In the inner loop that runs for many epochs, I sample k $(= 8)$ passages from the closed-set $(r = 100)$. This is fast because we are no longer retrieving from the entire corpus in the inner loop and also sufficient because the closed-set has a high recall. During the inner loop, I update the retrievers (both document and query encoders) and use the latest model parameters for computing the loss functions. A round results in trained models that are then used for the next round. I find that 2 rounds are often sufficient, with decreasing marginal utility from the third round onward.

Distributional repositioning before inference I approximate the expectation terms in ELBOLOSS by sampling k passages from the closed-set $Q_{\text{top-r}}(.|x, y)$, which provides better supervision than $P_\eta(z|x)$ and leads to faster training. However, the models only ever get exposed to passages from the $Q(.|x, y)$ distribution, which limits their ability to generalize over passages from $P_\eta(.|x)$ during inference. To remedy this, I instead sample passages from an α-mixture of the two distributions: with probability α, $z \sim P_\eta(.|x)$ and with probability $1 - \alpha$, $z \sim Q(.|x, y)$. In the initial rounds I set low values of α and increase it toward the end to reposition the passage distribution and better match with $P_\eta(.|x)$ at test time. The retriever and the generator can be trained by sampling passages from different α-mixtures and I utilize this to avoid retriever overfitting (with $\alpha = 1$) while maintaining generator groundedness (with $\alpha = 0.25, 0.5$).

Training individual models to convergence In practice, the retriever and the generator when jointly trained converge at different times. The single loss term in MARGINALIZEDLOSS (Equation 5.1) hides the convergence of individual models; one model starts to overfit while the other model still hasn't converged. With ELBOLOSS, there are two terms in Equation 5.2: the first connecting $Q(z|x, y)$ and $P_\theta(y|x, z)$, the second connecting $Q(z|x, y)$ and $P_\eta(z|x)$. After training for a few epochs, I freeze the guide and train the models independently until convergence based on their individual losses.

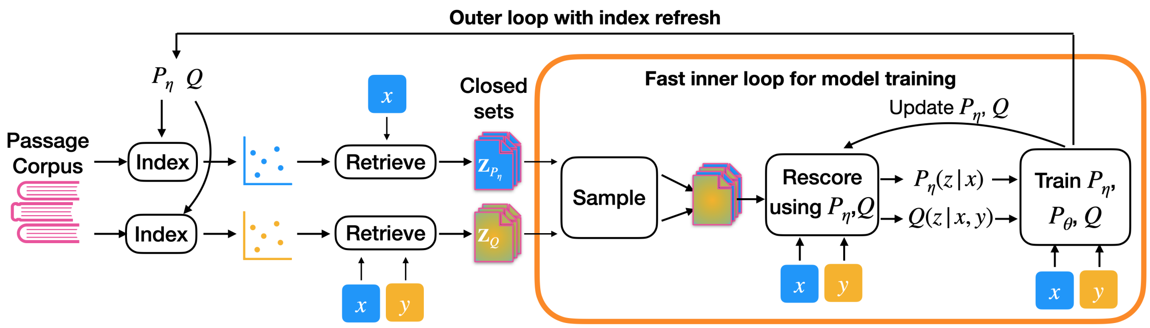


Figure 5.2: **An overview of iterative closed-set training.** I iterate through the outer-loop and call each execution a round. At the beginning of the round I re-index the passage corpus using the latest retriever $P_\eta(z|x)$ and guide-retriever $Q(z|x, y)$ to create a high-recall closed-set of top-r passages for each retriever and query. Then, in the fast inner loop, I train the models for multiple epochs by sampling passages from the fixed closed-set and recomputing the probability distributions. The trained models are then used in the next round.

5.4 Experimental Evaluation

I evaluate on two open-ended knowledge-intensive tasks: informative conversations and free-form question answering. I ask the following three research questions:

RQ$_1$ Relevance: Are the retrieved passages more relevant? (Section 5.4.4)

RQ$_2$ Groundedness: Does the generator make better use of the retrieved passages? (Section 5.4.5)

RQ$_3$ Generation Quality: Does this lead to better end-to-end performance? (Section 5.4.6)

5.4.1 Models

Retriever Models I model the retriever $P_\eta(z|x)$ and the guide-retriever $Q(z|x, y)$ using ColBERT (Khattab and Zaharia, 2020). ColBERT encodes the query tokens q_i and the document tokens d_j independently using BERT, normalizes to produce unit-vectors E_{q_i} and E_{d_j}, and defines similarity as $S_{q,d} = \sum_i \max_j E_{q_i}^T E_{d_j}$. Unlike DPR's [CLS] token embedding (Karpukhin et al., 2020), with ColBERT's late-interaction paradigm the query and document tokens retain their identities and contribute to a finer-grained term-wise similarity leading to state-of-the-art retrieval results on open-domain QA benchmarks (Khattab et al., 2021). To convert similarity scores into a probability distribution, calculate the softmax of the scores over the k sampled passages. For the posterior-retriever, I concatenate the input and the output into the query $q = [x\ y]$. ColBERT pre-trained on the MS-MARCO passage ranking dataset is widely used for other tasks and I use it for the Wizard of Wikipedia task. However, the MS-MARCO NLGen task contains queries from the passage ranking pre-training dataset.

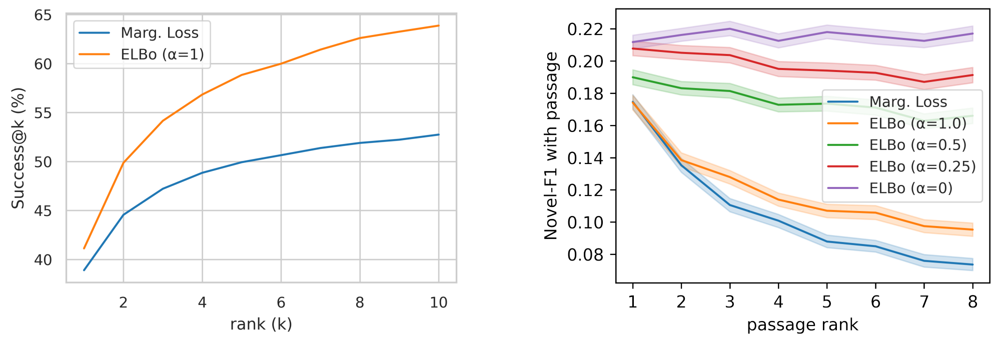


Figure 5.3: **Relevance and Groundedness of models trained on the Wizard of Wikipedia dataset.** (**left**) success@k of retrieved passages w.r.t. rank and (**right**) Novel-F1 between decoded output and retrieved passage w.r.t. retrieved passage rank. The ELBOLOSS retriever is more effective at retrieving the gold passage than the MARGINALIZEDLOSS retriever, especially when we consider the top-10 passages for this one-to-many task. The ELBOLOSS generators have higher overlap with top-k retrieved passages and the overlap increases as α decreases.

Therefore, following Lewis et al. (2020b), I use Natural Questions to pre-train ColBERT (Khattab et al., 2021) for the MS-MARCO NLGen task.

Generation Model Following Lewis et al. (2020b) I use a pre-trained BART model and fine-tune it for the respective tasks during training. It is conditioned on both the context and the document and trained to produce the target. At test time, I decode using beam-search with a beam size of 4.

5.4.2 Tasks

Informative conversations Informative conversations are open-ended because people have the agency to drive them in different directions at every turn (one-to-many) and are knowledge-intensive because the utterances contain specific bits of world knowledge. I evaluate with Wizard of Wikipedia (WoW) dataset (Dinan et al., 2019b), where an "apprentice" chats (via text) with a "wizard", being curious about different topics, and the "wizard" grounds their response in a sentence from Wikipedia. The input for this task is the conversational history x, the output is the wizard's utterance y and the models can retrieve individual passages z from all of Wikipedia (≈26 million passages). I use the version of this dataset provided in the KILT benchmark (Petroni et al., 2021) and report leaderboard performance on the held out test set. I use the dev set to answer the granular research questions.

	Wizard of Wikipedia				MS MARCO NLGen			
Method	MRR	S@1	S@5	S@10	MRR	S@1	S@5	S@10
Gold-sup.	45.2	35.6	57.0	63.1	28.9	19.5	40.4	47.7
Marg.	43.8	38.9	49.9	52.8	30.4	19.4	43.4	53.2
ELBo ($\alpha = 1$)	**49.0**	**41.1**	**58.8**	**63.9**	**32.1**	**21.2**	**45.3**	**54.4**

Table 5.1: **Relevance evaluation of trained retrievers.** Our method (ELBOLOSS Retriever, $\alpha = 1$) strongly improves over the baseline (MARGINALIZEDLOSS Retriever) for the one-to-many Wizard of Wikipedia dataset, in particular for $k = 5, 10$. The ELBo posterior finds z_{gold} with high success providing better supervision during training. (MRR = Mean Reciprocal Rank, Success@k both in percentages.)

Free-form Question Answering I use the MS-MARCO NLGen dataset (Nguyen et al., 2016) where the task is to generate natural-sounding answers to questions. This free-form open-QA task is one-to-one but more challenging than other *extractive* open-domain QA datasets. The dataset is a subset of MS-MARCO questions whose answers were reviewed by a separate editor and rewritten if they had a high overlap with one of the provided passages (indicating that the original editor may have copied the passage directly). These "well-formed answers" are meant to be complete sentences (such as can be read out by a conversational assistant) and are long (median length 11 words). The input for this task is a query x, the output is a well-formed answer y, and the models can retrieve from the MS-MARCO passage collection (8.8 million web passages). The public benchmark and the test set is no longer available for evaluation. Instead, I split the public validation set into a validation and test set and show results on the test set.

While both datasets annotate the passages referred to by the person who wrote the target output (gold passages), I only use them for evaluation and validation and not for training.

5.4.3 Baselines

Apart from the two main methods – MARGINALIZEDLOSS and ELBOLOSS – I train two additional baselines: **gold-supervised** and **generator-only**. For the Gold-supervised baseline, I assume that the gold-passage z_{gold} is available during training and train a retriever by maximizing its log-likelihood. I take random passages from ELBo Retriever's closed-set as negatives (excluding top-10 to avoid false negatives). I train the Gold-supervised generator by simply maximizing $P_\theta(y|x, z_{gold})$. For the generator-only baseline, I ignore the existence of passages and directly maximize $P(y|x)$ with a sequence to sequence model.

5.4.4 Relevance Evaluation

I evaluate the relevance of the retrieved passages (**RQ**$_1$) using the gold passage labels supplied by each dataset. For one-to-many tasks, we expect the label-relevant passage to be one of the top-k ($k = \{1, 5, 10\}$) retrieved passages. Thus, I report Success@k (S@k for short)[1], the percentage of inputs for which any gold provenance passage is retrieved within the top-k ($k = \{1, 5, 10\}$) passages. I also report Mean Reciprocal Rank (MRR), a common IR evaluation metric.

Our results are shown in Table 5.1. With Wizard of Wikipedia, ELBOLOSS retriever markedly outperforms MARGINALIZEDLOSS. Both systems have a relatively high Success@1 and easily handle 38.9–41.1% of the examples, but the ELBOLOSS retriever continues to find many more relevant passages at larger retrieval depths k widening the gap to 11 points for Success@10. With MS MARCO NLGen, ELBoLoss outperforms MarginalizedLoss by 1–2 points across our metrics, reflecting smaller—but nonetheless consistent—gains when compared with Wizard of Wikipedia, a one-to-many generation task. Based on manual inspection, I find many false negatives (corroborated by Arabzadeh et al. (2022)), i.e. passages that contain the answer but aren't marked as gold, leading to a lower Success@1 compared to Wizard of Wikipedia.

Effect of the guide-retriever MARGINALIZEDLOSS depends on the retriever $P_\eta(z|x)$ to find label-relevant passages during training and is therefore recall limited. MARGINALIZEDLOSS's success@100 on Wizard of Wikipedia saturates at 55.8% (not reported in the table) without much hope of further improvement because z_{gold} that are never retrieved cannot provide positive examples for supervision. With ELBOLOSS, the guide-retriever retrieves label-relevant passages with >85% success@10 (Table C.4) for both the datasets providing better supervision than MARGINALIZEDLOSS. Consequently, ELBOLOSS retriever's success@5 is higher than MARGINALIZEDLOSS despite containing 20× less passages and reaches 69.3% for success@100.

Comparison with Gold-supervised retriever I find that the Gold-sup. retriever quickly overfits during training resulting in a lower performance than using ELBo loss. I also see lower performance when training with ELBOLOSS while sampling purely from $Q(.|x, y)$ (i.e., $\alpha = 0$, Table C.3) because it is low-entropy and the same label-relevant passages get repeatedly sampled. By sampling from $P_\eta(.|x)$, the KL divergence is minimized over a wider and realistic support set of passages. Perhaps, there are many passages that have some label-relevant phrases making them partially relevant and $Q(z|x, y)$ "teaches" $P_\eta(z|x)$ to capture these phrase-level relative differences. Sampling from $P_\eta(.|x)$ leads to better generalization, has similarities to distillation and is an interesting direction for future work.

[1] With a single gold passage Recall@k and Success@k are numerically identical and sometimes used interchangeably; I prefer S@k because it is less ambiguous and widely used in the IR community since Voorhees (2004).

Dataset	Method	Top-1 F1	Top-1 Nov-F1	Max. of Top-5 F1	Max. of Top-5 Nov-F1
WoW	Gold-sup. Generator	18.84	17.12	32.47	31.92
	Marg. Generator	18.63	17.46	26.19	25.39
	ELBo Generator ($\alpha = 0.25$)	**21.34**	**20.78**	**34.16**	**34.24**
MSM	Gold-sup. Generator	36.13	28.34	49.29	43.45
	Marg. Generator	33.12	25.39	45.76	39.45
	ELBo Generator ($\alpha = 0.5$)	**34.52**	**26.49**	**46.91**	**40.47**

Table 5.2: **Groundedness evaluation of trained generators.** Our method ELBOLOSS ($\alpha = 0.25, 0.5$) shows more overlap between generated output and the retrieved passage than MARGINALIZEDLOSS and for the Wizard of Wikipedia dataset the gap increases as we consider the maximum over top-5 passages. (Novel-F1: discounts commonly occurring words and context words (x)).

Overall, I find that ELBOLOSS improves relevance of retrieved passages over MARGINALIZEDLOSS for two qualitatively different tasks, with larger gains for the one-to-many generation task.

5.4.5 Groundedness Evaluation

I now examine **RQ**$_2$, studying the degree to which the generator relies on the retrieved passages for producing its output. To quantify this *groundedness*, I compute F1-overlap between a *retrieved passage* (not necessarily the gold passage) and the produced text when generation is conditioned on that passage. I get the retrieved passages for each method (except generator-only) using the corresponding retriever.

As an analogue of Success@k, I propose *Max. F1@k*, the largest F1-overlap exhibited by any generated output with the corresponding retrieved passage fed to the generator. I also propose Novel-F1, a new metric that discounts words that occur frequently and words that already appear in the context x, since otherwise these tokens dominate raw F1 in practice (up to 80%, see Section C.6) but are not indicative of grounding.

Our results are shown in Table 5.2 and Figure 5.3. For the Wizard of Wikipedia dataset, I observe that our ELBOLOSS generator outperforms MARGINALIZEDLOSS by 2.7 F1 (14.5% relative improvement) and 3.3 Novel-F1 (19% r.i.) when given the top retrieved passage. For the MS MARCO NLGen dataset, I observe smaller but consistent gains in groundedness (1–2 F1, Novel-F1) with ELBOLOSS compared to MARGINALIZEDLOSS. In Figure 5.3 (right), MARGINALIZEDLOSS generator's overlap decays rapidly beyond the top passage, whereas the ELBOLOSS ($\alpha = 0.25$) generator's overlap declines gradually. This shows that the ELBOLOSS generator stays grounded beyond just the top passage, a desirable property in one-to-many generation systems. We also see (in Figure 5.3, right) that groundedness increases as α decreases. However, a generator trained with $\alpha = 0$, despite being maximally grounded (Appendix Table C.1), has lower end-to-end performance (Appendix Table C.2) because it is unduly "trusting" of the provided passage (nearly flat line in Figure 5.3) and does not abstain from using irrelevant passages.

We see the same effect with the Gold-supervised generator on MS-MARCO NLGen: it is more grounded but has lower downstream performance.

Overall, I find that ELBOLOSS improves grounding of the generator over MARGINALIZEDLOSS for two qualitatively different tasks, with larger gains for the one-to-many generation task.

5.4.6 End-to-end Evaluation

To evaluate the end-to-end quality of our systems, I calculate F1 and Novel-F1 (defined in Section 5.4.5) of the decoded output with the *human-written gold output*. To allow for the possibility of the generator using any part of the *gold passage* (and not just the human-written gold output) for the Wizard Of Wikipedia task, I use Knowledge-F1 (defined by Shuster et al. (2021)): F1 between the sampled generation and the *gold passage*. Since it is reasonable to expect the gold passage to be in the top-k for k>1 for one-to-many tasks (as in Section 5.4.5), I also compute the max. over top-k retrieved passages.

The results are summarized in Table 5.3. For the Wizard of Wikipedia dataset, using only the top retrieved passage ELBOLOSS leads to 6.7% relative improvement in Novel-F1@1. But in the one-to-many setting, the label-relevant passage is an arbitrary choice from amongst the context-relevant passages. I account for that using max. overlap over the top-5 passages and see larger improvements for ELBOLOSS, namely 1 F1, 2 Novel-F1 ($\sim$ 15% r.i.), and 1.5 K-F1 ($\sim$ 10% r.i.). For MS Marco NLGen, we see a small but consistent increase due to ELBOLOSS over MARGINALIZEDLOSS: 1.5 F1 and 2 Novel-F1 across passage depths.

I also submit the above models (ELBOLOSS and MARGINALIZEDLOSS) to the Wizard of Wikipedia task on the KILT leaderboard. ELBOLOSS consistently outperforms the baseline MARGINALIZEDLOSS across all metrics (see Table 5.4). The table also reports Recall@5, which evaluates retrieval at a coarser granularity, namely at the *full Wikipedia page* level, though so far I have investigated it directly at the passage level. Consistent with the results in Table 5.1, our method also outperforms MARGINALIZEDLOSS in retrieval metrics. In fact, our ELBOLOSS model achieves state-of-the-art performance across all the generation metrics (F1, ROUGE-L, KILT-F1, KILT-ROUGE-L) on the leaderboard, though it is not the strongest on R-Prec and Recall@5.[2]

To conclude, I have evaluated the ELBOLOSS and MARGINALIZEDLOSS using a one-to-one free-form QA dataset and a one-to-many dataset of informative conversations. Our results show that our method ELBOLOSS

[2] Earlier results on the KILT leaderboard for Wizard of Wikipedia should be interpreted with caution, as the KILT authors recently updated the train/dev splits due to anomalies in the preprocessing script. I have used the updated version for our model and baseline.

		Top-1			Max. of Top-5		
Dataset	Method	F1	N-F1	K-F1	F1	N-F1	K-F1
WoW	Gold-sup.	16.70	8.53	11.64	24.95	14.87	16.16
	Gen. Only	16.11	5.15	8.05	–	–	–
	Marg.	18.79	10.45	12.61	26.52	16.42	16.02
	ELBo	**18.86**	**11.12**	**13.08**	**27.56**	**18.67**	**17.69**
MSM	Gold-sup.	59.25	36.22	–	71.44	55.02	–
	Gen. Only	51.75	14.71	–	–	–	–
	Marg.	60.18	37.19	–	72.22	56.06	–
	ELBo	**61.46**	**39.65**	–	**73.18**	**58.19**	–

Table 5.3: **End-to-end automatic evaluation of the system consisting of a trained retriever and a trained generator.** Our method ELBOLOSS improves over MARGINALIZEDLOSS when considering Max. overlap of generated output with target output over top-5 passages for the Wizard of Wikipedia dataset and also for top-1 with MS Marco NLGen dataset. (Novel-F1: discounts commonly occurring words and context words (x), Knowledge-F1: overlap of generated output with gold passage.)

	R-Prec	Recall@5	F1	ROUGE-L	KILT-F1	KILT-ROUGE-L
Re2G (prev. best)	**60.10**	**79.98**	18.90	16.76	12.98	11.39
Marg.	53.94	68.12	18.11	16.21	11.78	10.47
ELBo (curr. best)	56.08	74.26	**19.19**†	**17.06**†	**13.39**†	**11.92**†

Table 5.4: **Wizard of Wikipedia KILT leaderboard evaluation.** ELBOLOSS achieves SoTA on generation metrics (F1, ROUGE-L, KILT-F1, KILT-ROUGE-L indicated with †) as of Oct 2021 and improves relevance over MARGINALIZEDLOSS

trains a better retriever, a more grounded generator and improves end-to-end performance, especially in the one-to-many setting.

5.5 Discussion

Hallucination, grounding and correctness Shuster et al. (2021) show that providing retrieved passages to a generator reduces hallucination. Our work increases grounding in the retrieved passage, promising to further reduce hallucination. Even though the generator is now more likely to use content from the provided passage (rather than hallucinating from parametric memory), that does not guarantee *correctness*. Our token-level overlap metrics that evaluate grounding do not capture this aspect either. There is scope for future work to address this gap with better training methods (and evaluation metrics) that produce (and reward) grounded and correct outputs.

Practical matters: Trust and control For tasks like QA, we want a “conservative” generator: it should abstain from using a passage that doesn’t contain the answer. For more open-ended tasks like informative conversations, we want the generator make use of diverse passages. In this chapter, I show that by reducing α, the distribution of the passages shifts from $P_\eta(.|x)$ to $Q(.|x,y)$ and the generator increasingly trusts the retrieved passages (Figure 5.3). System designers can use the α-mixture as a tool to modulate the degree of trust placed by the generator in the retrieved passages. Further, when a “trusting” generator is deployed in real-life settings (e.g., in open-domain socialbots like Chirpy Cardinal from Chapter 3), external business logic can select an appropriate passage from the top-k retrieved passages and effectively control the generated content.

Comparison with Fusion-in-Decoder For QA style tasks, the top-1 passage may not contain the answer, and it is useful to look for the answer in the top-k passages. If the generator can attend to the top-k passages simultaneously, it has the opportunity to synthesize all of them and produce the correct answer more often. In the Fusion-in-Decoder (FiD) architecture (Izacard and Grave, 2021b), the decoder has a cross-attention mechanism over multiple passages helping it “gather the evidence” from many passages at once. However, in my experiments, I found marginal improvements while using Fusion-in-decoder. I attribute this to the combination of two reasons: the one-to-many generation setting and the lack of synthesis in our training data. Unlike QA style tasks, where many passages add evidence for the same answer, in the one-to-many generation setting, each new passage is adding diverse new information. For example, in Figure 5.1, each passage is adding new information about different jazz musician rather than providing evidence for talking about “Louis Armstrong”. Furthermore, the labels y in the training data do not synthesize multiple passages. They were collected by asking crowd workers to base their answer on one passage at a time, limiting the utility of the FiD architecture. If datasets in the future synthesize information from multiple passages, the FiD architecture might show improvement over the current decoder architecture.

In the Fusion-in-Decoder Knowledge Distillation (FiD-KD) training method (Izacard and Grave, 2021a), the decoder’s cross-attention weights corresponding to each passage can be used for relevance supervision. In EMDR^2, an end-to-end trained multi-document QA system, Sachan et al. (2021) use FiD architecture for the reader. However, unlike FiD-KD, instead of training the retriever using the FiD reader’s attention weights, they feed passages individually to the reader and use MARGINALIZEDLOSS for updating the retriever parameters. They claim their approach is superior to FiD-KD, because EMDR^2 only requires one cycle of end-to-end training whereas FiD-KD requires multiple cycles and EMDR^2 more robust to retriever initialization compared to FiD-KD. Thus, it is unclear if FiD-KD training can provide good relevance supervision and I believe our approach of using a posterior model while conditioning on individual passages is more precise and effective.

Inductive biases of HINDSIGHT ELBOLOSS is a better approximation of $P(y|x)$ than MARGINALIZEDLOSS because, during training, ELBOLOSS samples k passages from Q that are more label-relevant. As described in Section 5.3, the ELBOLOSS also encourages two inductive biases. First, the generator's loss for a passage is weighted by its label-relevance as quantified by the posterior distribution. This makes it more grounded. Second, the reverse KL-divergence acts like a soft implication. The retriever needs to assign a high probability mass to label-relevant passages and is penalized less severely for placing probability mass on context-relevant passages. But, in the hypothetical limit, where I marginalize over all passages, a learned $P_\eta(z|x)$ that maximizes ELBOLOSS (a lower bound) will also maximize MARGINALIZEDLOSS. *So does this difference arise only because we are limited to sampling k passages during training?* Even though there is no way to empirically answer this question, based on the derivation of ELBOLOSS, I argue otherwise. The inequality in ELBOLOSS is due to $D_{\text{KL}}\Big[Q(z|x,y)\|P(z|x,y)\Big] \geq 0$ used in the derivation (see Appendix C.1), where $P(z|x,y)$ is the true posterior and $Q(z|x,y)$ is an approximation of it. In fact, ELBOLOSS is exactly equal to $P(y|z) - D_{\text{KL}}\Big[Q(z|x,y)\|P(z|x,y)\Big]$ and is therefore maximizing not just $P(y|z)$ but also minimizing $D_{\text{KL}}\Big[Q(z|x,y)\|P(z|x,y)\Big]$. Thus, in the limit, even if we were able to compute the expectations exactly, ELBOLOSS optimizes for a slightly different objective; the larger the value of $D_{\text{KL}}\Big[Q(z|x,y)\|P(z|x,y)\Big]$, the bigger the difference. In HINDSIGHT, the representational capacity of Q is limited to that of $P_\eta(z|x)$, because I use the same architecture for both Q and $P_\eta(z|x)$, with the only difference being that Q has access to privileged information y. I hypothesize that this makes the second term, $D_{\text{KL}}\Big[Q(z|x,y)\|P(z|x,y)\Big]$, act like a regularizer. This idea is taken even further in Section 5.3, where I introduce the idea of distributional repositioning. Here, I sample passages from the α-mixture, inducing a bias toward the test distribution and improving generalization, but further loosening the theoretical bound. In this chapter, I show how it can be fruitful to introduce desirable inductive biases even if it means we loosen mathematical bounds.

5.6 Conclusion

In this chapter, I proposed HINDSIGHT, a system that introduces a guide-retriever to improve supervision for both the retriever and the generator for retrieval-augmented, open-ended generation. During training, the guide retriever uses the target output of each example in order to find relevant passages, leading to better retrieval and more grounded generation. The resulting system achieves considerable empirical improvements over existing work, improving retrieval quality by up to 23%, grounding by up to 19%, and end-to-end output quality by up to 6.4%.

Chapter 6

Conclusion

In this dissertation, I started with the goal of building neural systems for informative conversations that can have in-depth conversations on a broad range of topics. I situated my work in Chapter 1 and gave an overview of related work in Chapter 2. After that, in Chapter 3, I demonstrated how to build a social chatbot and used that as a platform to answer **RQ**$_1$: **"What are the important problems for informative conversations?"**. One of the problems was our lack of fine-grained understanding of what makes a good informative conversation. So, in Chapter 4, I study human-human informative conversations and answer **RQ**$_2$ **"What strategies do humans employ when talking informatively with other humans?"**. I also present a case study where I select generated responses with better acknowledgements using conditional mutual information. To fix issues with other strategies, e.g. transition and detail selection, I needed to change how the retriever found knowledge snippets in the first place. In Chapter 5, I provide a method for posterior-guided training of retrievers for open-ended generation. I refer the reader to Section 1.5 for insights on my research process and the lessons it carries for future researchers.

In the rest of this chapter, I highlight the limitations of my work, ways in which open-domain dialogue systems can be improved and provide a sneak peek into future research directions.

Richer spoken dialogue: In this dissertation, we limited ourselves to building systems for text-based dialogue. The de facto method to allow audio interactions is to first use an Automatic Speech Recognition (ASR) system to convert a user's audio to text, then use the text to generate a textual response and then use a Text to Speech (TTS) system to convert it back to audio which is played back to the user. This is how the Alexa Prize Socialbots were deployed as well.

Many nuances such as emotion and emphasis are lost in the conversion. For example, imagine the question

"Why did the president visit the war zone?" being said with an emphasis on different words that changes the implied question.

- "Why did *the president* visit the war zone?" would imply why the president? Why not someone else?
- "Why did the president *visit* the war zone?" would question the act of visiting. Could he have done something else like called in, or send aid?
- "Why did the president visit *the war zone*?" would question the location of his visit. Perhaps the context of the situation calls for asking if he could have visited some place else.

The answer depends on the verbal emphasis and is therefore important to capture in any spoken informative system. But more generally any interface meant to be deployed for spoken usage needs to capture these nuances.

In certain cases, this form of mechanical turn taking is confusing at best and debilitating at worst. I observed users getting frustrated with our Chirpy Cardinal system when deployed via Alexa devices. The system would ask them a question that required them to think and when they paused to think it would end the turn. Naturalistic turn-taking is a "low-cost" behavior for human-human conversations. If a participant starts their turn by mistake, it is easy to stop, recover and continue later. It is easier to speak for a long duration because the speaker can feel confident that the participant can interject them if there is any confusion. These turn-taking mechanisms that we (as humans) are used to, are completely missing from deployed systems. As a first step toward fixing it, Li et al. (2022b) predicts the places where the current listener could start speaking. Using this information, a dialogue agent can prepare a partial response ahead of time and start speaking as soon as the user ends their turn. But more work needs to be done to enable truly naturalistic turn-taking, including but not limited to on-device audio processing (to cut down latency) and a dialogue agent designed with these kinds of interjections in mind.

Long-term consistency: So far, in this dissertation, there was no explicit goal-oriented planning involved in response construction. While the goals are not crystal clear for open-ended dialogue (as opposed to task-oriented), there are meta-goals that we can derive from psycholinguistics. The desire to appear knowledgeable, save face, be factually correct, appear concerned and empathetic, etc., can be goals that dialogue agents can actively optimize for during the conversation.

Prior work (Li et al., 2016c; Williams and Zweig, 2016) has attempted to use Reinforcement Learning (RL) for dialogue generation. But there have been many advances in the past few years that can further improve the applications of RL to dialouge. For instance, large language models can act as simulators that play out possible conversations from a given utterance. Using this as feedback, a dialogue model can learn if the local choices (i.e. the current response) leads toward globally favorable outcomes.

Though I touch on a few specific issues above, there are many avenues for improving NLP systems for informative conversations. Through this thesis I provide a foundation for informative neural systems; ideas and code that future researchers can build upon. Many problems are still unsolved and there are doubtless many new ideas that will push the field forward.

Appendix A

Chirpy Cardinal

A.1 Additional RGs

A.1.1 Coronavirus

Our bot has been available to users since early December 2019 – thus covering the entire period of the COVID-19 pandemic so far. As the pandemic progressed, users requested to talk about the pandemic more and more, and showed more interest in discussing home life rather than outside activities. We created the Coronavirus RG to meet users' expectations that our bot should be fully aware of the unfolding situation, on both an informational and emotional level.

The RG is activated when the user mentions or requests to talk about the Coronavirus. To be mindful of the stress that the pandemic has placed on many people, we never initiate the topic ourselves. We then give a comforting message acknowledging the emotional effect of the pandemic, and give the user the option to hear up to 4 news updates. After that, we ask an uplifting open-ended question, and handle the user's response by generating a response from our GPT-2 model (Section 3.7). We then perform a scripted transition to the Movies RG. For example after we've read out a news headline:

> **User**: *No thank you. I'm exhausted from listening to the news all the time.*
> **Bot**: *No problem. I know this situation has been very stressful for many people; I've read online that it can be helpful to remind ourselves of small, good things that have happened to us lately. What's something nice that's happened to you in April?*
> **User**: *Well it was my birthday!*
> **Bot**: *That's great. Did you have fun?*

...

A.1.2 News

The News RG provides the user with news stories that interest them, and discusses them. We set up a cron job to query the Washington Post API[1] and scrape several news-based subreddits[2] every day, and place the data in an ElasticSearch index. When the user asks to talk about news, the News RG asks the user for a topic (e.g., *Joe Biden*), if it wasn't already supplied. The RG then queries the ElasticSearch index for news stories with the desired topic in the headline, selects the most recent one, reads out the headline to the user, and asks if they'd like to hear more. If accepted, we read out the first three sentences of the article.

Our original goal was to allow the user to ask follow-on questions about the article, and to answer them with a **Neural Question Answering** model. We hoped this would help realize our design goals of conversational phrasing and enabling user initiative (Section 3.1). To begin this process, the News RG would invite the user to ask questions. We then used the SpaCy coreference resolution module (Honnibal and Montani, 2017) to decontextualize the user's question with respect to the last two utterances from the News RG. For example, *how many votes did he win?* might be transformed to *how many votes did Joe Biden win?* The decontextualized question, along with the entire news article, was then sent to a BERT-Large model (Devlin et al., 2019) trained on the Stanford Question Answering 2.0 dataset (Rajpurkar et al., 2018) by HuggingFace.[3] The model would output either a span in the article, or 'no-answer' – meaning the question cannot be answered by the provided article.[4]

Unfortunately, in our internal testing, we found that this system had several substantial problems. First, errors in the coreference module were common, and would cascade to the QA module. Second, we found that users asked a very different distribution of questions, compared to the SQuAD training questions. For example, users were likely to ask more open-ended or causal questions (e.g., *what happened next?*, *why did they do that?*). These are difficult for off-the-shelf QA models, which tend to excel in answering factoid-style questions. Third, users were likely to ask questions whose answers are not present in the news article. Though our model was trained on SQuAD 2.0 (which contains unanswerable questions), it would often choose an irrelevant answer that type-checks with the question, as Jia and Liang (2017) have also reported. Even when the QA model correctly classified unanswerable questions, we would have needed to build a substantial open-domain question answering system to handle these questions. Overall, these problems made our system

[1] An API call to scrape Washington Post news articles provided by Amazon Alexa.
[2] /r/News, /r/Sports, /r/Politics, /r/Futurology, /r/Science, /r/Technology, /r/WorldNews
[3] https://github.com/huggingface/transformers
[4] Since the article was often much larger than the maximum context size for BERT, we ran the model on chunks. Within each chunk, we discarded spans which were ranked lower than 'no-answer', then merged the answers and re-ranked by confidence of the predictions.

a poor and unreliable user experience; requiring more time and effort to fix than we had available.

A.1.3 Other RGs

Launch Handles the first few turns of the conversation (introducing the bot and learning the user's name). An example can be seen in Table 3.1.

Acknowledgement When the user changes topic to a new entity, this RG uses the entity's membership in certain Wikidata categories to select a one-turn scripted acknowledgement (e.g. *Oh yeah, I read ENTITY last year - I couldn't put it down!* if the entity is a book). This RG aims to give a natural and conversational acknowledgement that a new topic has been raised, before handing over to another RG (e.g. Wiki/Opinion/News) to discuss the entity in more depth.

Alexa Commands Users often try to issue non-socialbot commands (such as playing music or adjusting smart home devices) to our socialbot. This RG detects such commands, informs the user that they're talking to a socialbot, and reminds them how they can exit.

Closing Confirmation Our bot stops the conversation when the user issues a command like *stop* or *exit*. However, users indicate a possible desire to exit through many other more ambiguous phrases (e.g., *do you just keep talking*, *what's happening*). This RG detects such cases using the *closing* dialogue act label (Section 3.5.2) and regex templates, asks the user if they'd like to exit, and stops the conversation if so.

Complaint Provides an appropriate response when a user complaint is detected. This RG uses the Dialogue Act classifier's *complaint* label to detect generic complaints, and regular expressions to detect misheard complaints (the user saying that Alexa misheard them), clarification complaints (the user saying that Alexa is not being clear), repetition complaints (the user saying that Alexa is repeating itself), and privacy complaints (the user saying that they don't want to share information). We wrote different responses for each type of complaint, to reflect understanding of the user's concerns.

Fallback Always provides a response (*Sorry, I'm not sure how to answer that*) or prompt (*So, what are you interested in?*) to be used when no other RG provides one.

One-Turn Scripted Responses Provides handwritten responses to common user utterances (e.g. *help*, *chat with me*, *hello*) that can be handled in a single turn.

Red Question Detects if the user asks our bot a 'red question' – i.e., a question we are not permitted to answer, such as medical, legal, or financial advice – and informs the user that we cannot answer. To recognize these questions, we trained a multinomial logistic regression model on bag-of-words features, using data from the /r/AskDoctor, /r/financial_advice, and /r/LegalAdvice subreddits.

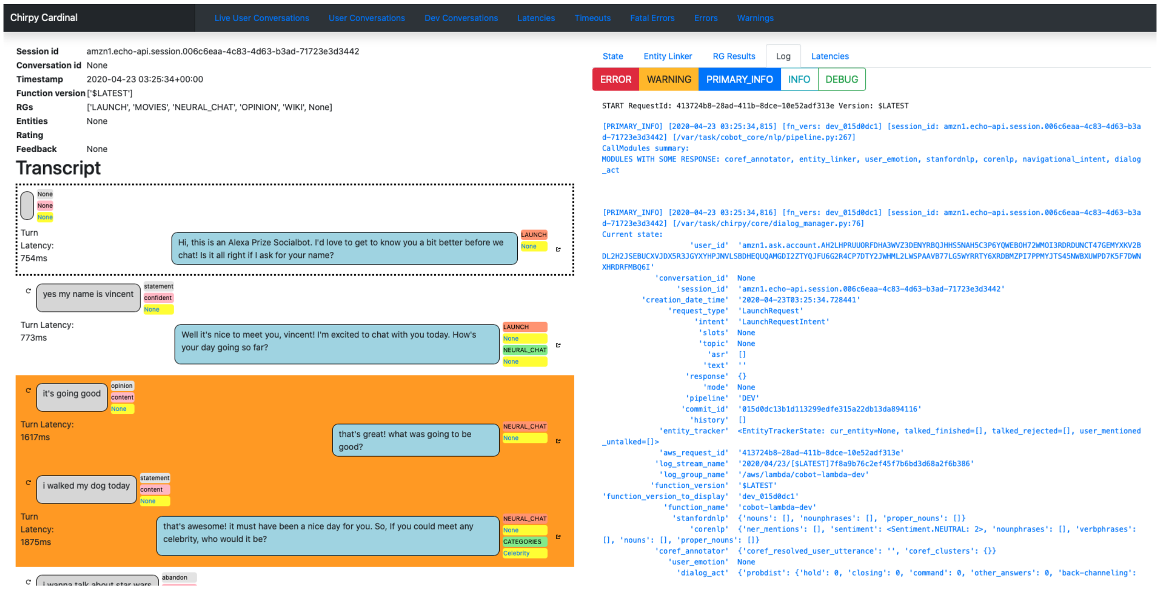


Figure A.1: **Screenshot of an example conversation in the dashboard.** The tags next to each utterance are annotations from the bot. The background color of the utterance is the latency of that specific turn (white being normal and orange being slow). The pane on the right shows the logs for the turn. Note that is not a conversation with a real Alexa User.

A.2 Tooling and Processes

A.2.1 Dashboard

We built a browser-based dashboard to provide ourselves with easy readable access to conversations and the associated metadata. The landing page shows aggregate rating statistics broken down by date and code version. The dashboard can filter conversations based on metadata such as number of turns, ratings, entities and RGs used. For each conversation, the dashboard displays important turn-level attributes, such as latency, entities, annotations, state information, RG results, and logs. It can provide a link pointing to a specific turn, which is very useful for discussions and issue tracking. The dashboard can rerun the conversation with the current version of our bot, to quickly test if our local changes fixed the problem. Aside from displaying conversations, the dashboard also has tabs to track errors and latencies, divided by severity level. Easy accessibility and visibility of errors made us more aware and likely to fix these errors quickly.

A.2.2 Processes

Code Review We realized early on that maintaining high code quality is important for maintainability and extensibility. We set up a circular code review process to ensure that any code we write is understandable by another team member and adheres to certain quality standards.

Integration Tests We also instituted integration tests, to ensure that our bot maintains certain core functionality. We often found that some changes we made in one part of the bot had unexpected and damaging effects in another part of the bot; integration tests helped to catch these issues.

Canary Testing We had two versions of our bot – **mainline**, which handled real customers, and **dev**, which we used for developing new features. At first, new dev versions were solely tested by team members, before being pushed to mainline. However, especially as the complexity of the bot grew, this method became insufficient to identify problems in new dev versions – meaning that bugs were being discovered in mainline. We set up a canary testing framework, which directs a controllable percentage (typically 10%-50%) of customer traffic to dev. This was very useful in allowing us to tentatively test out new features with larger numbers of people, before deploying to all customers, thus protecting our ratings.

UX Officer Each week, we have a dedicated UX officer, whose primary responsibility is to monitor the conversations, identify problems, and get a sense of the strengths and weaknesses of the current system. This person is also responsible for alerting other team members to things that need to be fixed, and communicating their overall findings to the rest of the team at the weekly meeting. The role rotates every week, so every team member has a chance to see the bot in action, and stay in touch with the overall user experience.

Sprint Planning and Issue Tracking We use Jira to track issues to be fixed – each is assigned to the person in charge of the relevant component. We have a weekly sprint planning meeting where we prioritize the most important things to work on over the next week, and use Jira to track the sprint.

A.3 Dialogue Act Classifier

A.3.1 Modifications to Label Space

We modified this schema to better fit the needs of our bot, adopting 19 out of 23 dialogue act labels from MIDAS paper, and creating 5 new labels: *correction*, *clarification*, *uncertain*, *non-compliant*, and *personal question* to support UX-enhancement features such as the ability to respond to clarifying questions. We dropped the labels *apology*, *apology-response*, *other*, and *thanks* since there were very few ($n \leq 80$) examples of them in the original dataset and we rarely observed these dialogue acts in our bot.

A.3.2 Labeling Procedure

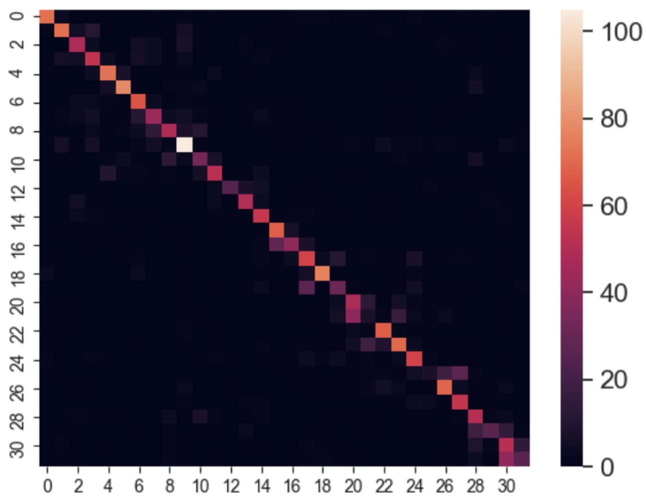
To create our gold-labeled dataset from our bot, we first determined which classes we most wanted to improve, based on per-class F1-Score for the baseline model and the new features we wanted to build. For example, since we wanted to improve our complaint handling, we prioritized this category. Next, we ran the baseline model on data from our bot to collect pseudo-labels. We randomly sampled 300 examples per label and then annotated whether the true label matched the predicted label. If not, we annotated what the correct label was. Using the pseudo-labels as a starting point increased efficiency, since the binary decision of "correct or incorrect" is much easier than the choice between 24 labels, and this method significantly reduced the number of non-binary decisions necessary. It also improved balance over classes, since it gave us greater control over the classes in the sample, and allowed us to prioritize certain categories. The result of training with gold-labeled examples is reported in Table 3.4.

A.4 Emotion classifier and analysis

In order to understand and analyze users' emotions, we finetuned a RoBERTa model (Liu et al., 2019; Wolf et al., 2019a) on the EmpatheticDialogues dataset (Rashkin et al., 2019), which contains 24,850 examples broken into an 80-10-10 train-dev-test split. In particular, our training and test data consisted of the first utterance from each dialogue (as it is the only one with a label), along with its label (one of 32 fine-grained emotions, listed in Figure A.2).

The RoBERTa model achieves a top-1 accuracy of 61.5% and an F1-score of 0.596. However, many of the misclassifications are due to the model choosing a label very similar to the gold label. For example, in the confusion matrix in Figure A.2, we see that *angry* is often misclassified as *furious*, and *terrified* as *afraid*, among others. In contrast, the top-5 accuracy is 92%.

One difficulty in applying this classifier to our user utterances is domain shift. The EmpatheticDialogues training utterances all describe a strongly emotional personal situation in complete written sentences, in a self-contained way (i.e., with no preceding context) – for example, *A recent job interview that I had made me feel very anxious because I felt like I didn't come prepared.* By contrast our user utterances are spoken, typically not complete sentences, require conversational context to understand, and encompass many different dialogue functions (such as giving commands, answering questions, choosing topics, greeting and closing, etc.). Importantly, most utterances are emotionally neutral. As the classifier has no 'neutral' label, it assigns spurious emotions to these neutral utterances.



Emotions Key			
0	jealous	**16**	sentimental
1	impressed	**17**	sad
2	grateful	**18**	lonely
3	proud	**19**	devastated
4	confident	**20**	furious
5	prepared	**21**	angry
6	content	**22**	disgusted
7	joyful	**23**	annoyed
8	excited	**24**	disappointed
9	surprised	**25**	ashamed
10	anticipating	**26**	embarrassed
11	hopeful	**27**	guilty
12	faithful	**28**	anxious
13	trusting	**29**	apprehensive
14	caring	**30**	afraid
15	nostalgic	**31**	terrified

Figure A.2: Confusion matrix for RoBERTa emotion classifier.

A.4.1 Relationship between Rating and User Emotion

To understand users' emotions and how they relate to our bot's performance, we replicated our experiment for dialogue act labels by applying a regression analysis, to the emotion classifier labels and the ultimate rating of each conversation.

Before performing this analysis, we removed all one-word utterances, since we assumed that these would not contain any emotion, and 66 common utterances that accounted for 40% of responses (e.g. *yes* and *no*), assuming that they were also neutral.

Figure 3.6 shows that, as we would expect, positive emotions have the largest positive coefficients and negative emotions have the largest negative ones. A possible explanation for the anomalies (e.g. "terrified" having a relatively large positive coefficient) is that the emotion classifier strongly associates certain entities with emotions and struggles to recognize when these entities are used in different contexts. For example, it associates "tiger" with "terrified", even when "tiger" is in a positive context such as "I like tigers."

A.5 Offensive User Experiment Details

A.5.1 Offense Type Detection

To determine the offense type, we hand-labeled 500 most common offensive utterances, which accounted for 53% of all the offensive utterances we collected to the date. We used 6 categories: sexual, insult, criticism, inappropriate topic, bodily harm and error. To classify the user utterance into one of these categories, we built regular expressions checking if the given user utterance contains one of the hand-labeled examples for an offense type. We then used the offense type to contextualize our `COUNTER+PROMPT` and `EMPATHETIC+PROMPT` responses.

A.5.2 Response Strategy Configurations

This section gives a detailed description of the configurations used in the Offensive User experiments (Section 3.6.7).

1. `WHY`: We ask the user why they made the offensive utterance (and this forms the entire bot utterance for the turn). The Offensive User RG responds with *OK* to whatever the user says next, then hands over to another RG to supply a prompt. For example: **Bot**: *Why did you say that?*, **User**: *because you weren't understanding me*, **Bot**: *OK. So, who's your favorite musician?*

2. `WHY+NAME`: Same as `WHY`, but we append the user's name to the end of the bot utterance. For example: *Why did you say that, Peter?*

3. `AVOIDANCE`: The bot politely avoids talking about the offensive topic, e.g. *I'd rather not talk about that.* This forms the entire utterance for the turn; the bot does not give any prompt to steer the conversation in a different direction.

4. `AVOIDANCE+NAME`: Same as `AVOIDANCE`, but we append the user's name to the bot utterance. For example: *I'd rather not talk about that Peter.*

5. `AVOIDANCE+PROMPT`: Same as `AVOIDANCE`, but we also give a prompt to change the topic. For example: *I'd rather not talk about that. So, who's your favorite musician?*

6. `AVOIDANCE+NAME+PROMPT`: Same as `AVOIDANCE+NAME`, but append a prompt to the end of the utterance. For example: *I'd rather not talk about that, Peter. So, who's your favorite musician?*

7. COUNTER+PROMPT: Strategy suggested by Brahnam (2005) and evaluated by Chin et al. (2020). In our interpretation of the strategy, we point out the inappropriate nature of the user utterance to the user, and attempt to move on to a different topic. For example, *That is a very suggestive thing to say. I don't think we should be talking about that. Let's move on. So, who's your favorite musician?*

8. EMPATHETIC+PROMPT: Strategy suggested by Chin et al. (2020) as the most appropriate strategy. In our interpretation of the strategy, we empathize with the user's desire to talk about inappropriate topics, and attempt to move on to a different topic. For example, *If I could talk about it I would, but I really couldn't. Sorry to disappoint. So, who's your favorite musician?*

For COUNTER+PROMPT and EMPATHETIC+PROMPT, unlike Brahnam (2005) and Chin et al. (2020), we handwrote responses tailored for the type of the user offense (e.g. sexual, curse, etc.)

Note that WHY and AVOIDANCE cannot be combined because they are contradictory, and WHY and CHANGE cannot be combined because we cannot ask two questions in one turn.

A.6 Opinion Agreement Policy Details

Here we will show demonstrations of the 3 different Opinion Policies. All the demonstrations below are generated by researchers and are not actual user conversations.

A.6.1 ALWAYS_AGREE Policy

We begin by asking the user their opinion on the entity. Then, we agree with the user on their sentiment on the entity and share our reason, and ask the user their reason for liking/disliking the entity. After that, we propose another reason for liking/disliking the entity, and ask whether the user agrees with our reason or not. Finally, we switch to a related entity and repeat the same steps as described before. A demonstration of the policy is shown below.

User: *let's talk about cats*

Bot: *Ok! Do you like cats?*

User: *cats are my favorite*

Bot: *Sounds like you like cats. Me too! I feel like cats are so fluffy. What about you?*

User: *i like cats because they are very independent*

Bot: *That's so true. That reminds me of another reason I love cats. I feel like they treat you as an equal. Do you agree?*

User: *i totally agree*

Bot: *What about dogs? Do you like dogs?*

User: *yes*

Bot: *Me too! You know, I think the reason I'm a fan of dogs is because they love you back no matter what. What do you think?*

User: *yeah i feel the same way*

Bot: *Totally. I also like dogs because of how nifty they are. Do you feel the same way?*

User: *Yeah you are right they are pretty nifty.*

Bot: *Thanks for sharing! It's nice to know your likes and dislikes. Do you want to know more about dogs?*

A.6.2 LISTEN_FIRST_DISAGREE Policy

We initially ask the user for their opinion and reason (i.e. "What do you think about cats?" followed by "Why do you like cats?"), then we disagree with the user by giving our opinion and reason. After that, we switch to a related entity and agree with the user on their sentiment on the new entity. A demonstration of the policy is shown below.

User: *let's talk about cats*

Bot: *Ok! Do you like cats?*

User: *cats are my favorite*

Bot: *What's your favorite thing about cats?*

User: *i like cats because they are very independent*

Bot: *That make sense. I have to be honest though, I'm not a big fan of cats actually. I feel like they don't respect my personal space. Can we agree on that?*

User: *yeah i guess you are right on that one*

Bot: *What about dogs? Do you like dogs?*

User: *yes*

Bot: *Me too! You know, I think the reason I'm a fan of dogs is because they love you back no matter what. What do you think?*

User: *yeah i feel the same way*

Bot: *Totally. I also like dogs because of how nifty they are. Do you feel the same way?*

User: *Yeah you are right they are pretty nifty.*

Bot: *Thanks for sharing! It's nice to know your likes and dislikes. Do you want to know more* *about dogs?*

A.6.3 CONVINCED_AGREE Policy

This is a stronger version of the previous policy in which we immediately disagree with the user and give our opinion and reason (i.e. "What do you think about cats?" followed by "That's good to know, but I don't quite get the appeal of cats. I feel like they are too independent. But I want to hear why you like cats"). After receiving an answer from the user, we agree with the user and appear convinced. After that, we switch to a related entity and agree with the user on their sentiment on the new entity. A demonstration of the policy is shown below.

User: *let's talk about cats*

Bot: *Ok! Do you like cats?*

User: *cats are my favorite*

Bot: *Glad to meet a fan of cats! I have to be honest though, I'm not a big fan of cats actually. I feel like they don't respect my personal space. But I'm interested to hear why you like cats?*

User: *i don't agree i like that about cats because that's how they show their love*

Bot: *That make sense. Now that I think about it, there are a few things I like about cats. For example, they are very independent. What do you think?*

User: *yeah that's right.*

Bot: *What about dogs? Do you like dogs?*

User: *yes*

Bot: *Me too! You know, I think the reason I'm a fan of dogs is because they love you back no matter what. What do you think?*

User: *yeah i feel the same way*

Bot: *Totally. I also like dogs because of how nifty they are. Do you feel the same way?*

User: *Yeah you are right they are pretty nifty.*

Bot: *Thanks for sharing! It's nice to know your likes and dislikes. Do you want to know more about dogs?*

Appendix B

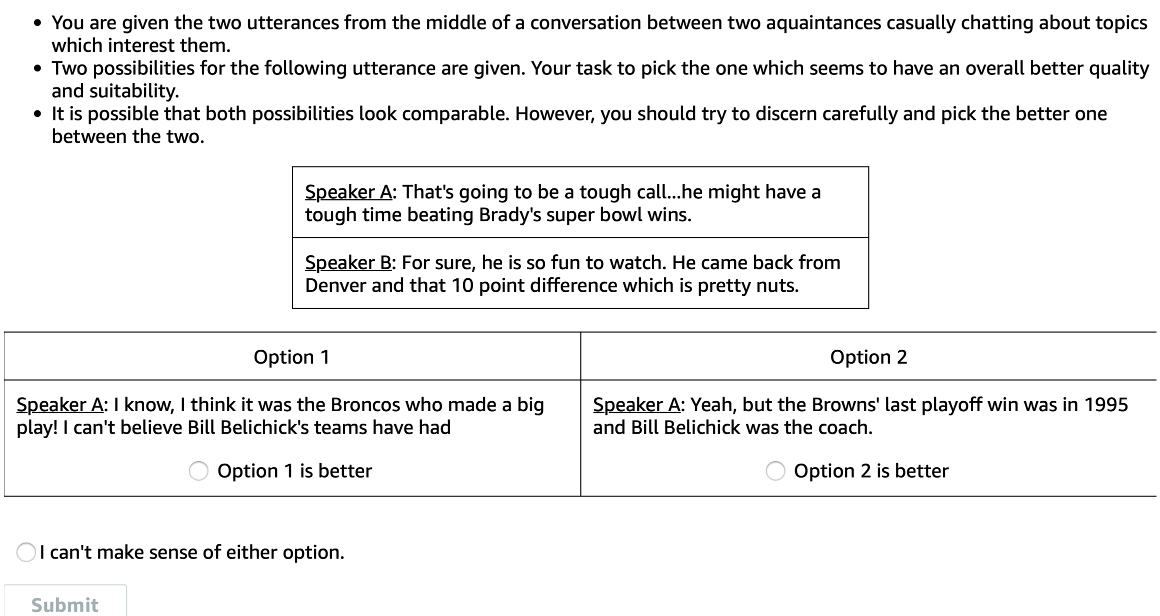
Improving acknowledgements: Experimental details

B.1 Model training details

Each model (main and ablation) was trained on a single NVIDIA Titan Xp GPU for 5 epochs and took approximately 8 hours to train. The training dataset had 51407 instances, validation 2491 and test 2728. The Topical Chat dataset and Switchboard corpus are in English language. The main model used for response generation had a validation loss (average negative log liklihood) of 2.05 which it reached after 2 epochs.

B.2 Annotation Details

We had 9, 19 and 19 unique annotators for experiments 1, 2 and 3 respectively. All three annotators agreed in 32/87 instances for experiment 1, 52/87 instances for experiment 2 and 32/99 instances for experiment 3.



- You are given the two utterances from the middle of a conversation between two aquaintances casually chatting about topics which interest them.
- Two possibilities for the following utterance are given. Your task to pick the one which seems to have an overall better quality and suitability.
- It is possible that both possibilities look comparable. However, you should try to discern carefully and pick the better one between the two.

Speaker A: That's going to be a tough call...he might have a tough time beating Brady's super bowl wins.
Speaker B: For sure, he is so fun to watch. He came back from Denver and that 10 point difference which is pretty nuts.

Option 1	Option 2
Speaker A: I know, I think it was the Broncos who made a big play! I can't believe Bill Belichick's teams have had ◯ Option 1 is better	Speaker A: Yeah, but the Browns' last playoff win was in 1995 and Bill Belichick was the coach. ◯ Option 2 is better

◯ I can't make sense of either option.

Submit

Figure B.1: Annotation interface for Best PMI v/s rest

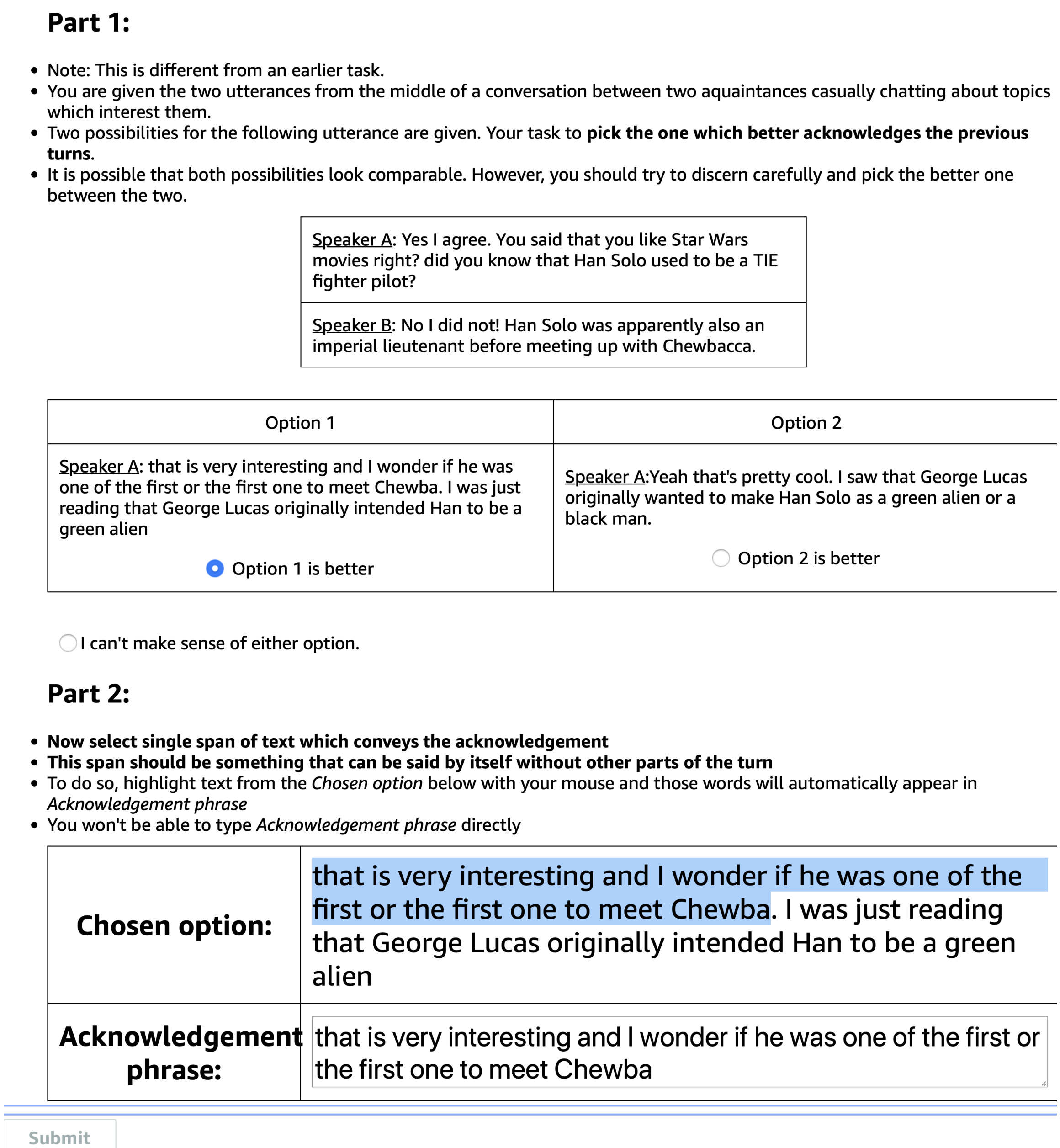


Figure B.2: Annotation interface for acknowledgement differences due to pcmi_h

Appendix C

Hindsight analysis

C.1 Derivation of ELBo Loss

Consider the KL divergence $D_{\text{KL}}\Big[Q(Z|X,Y)\|P(Z|X,Y)\Big]$ between true posterior $P(Z|X,Y)$ and our approximate model of the posterior $Q(Z|X,Y)$. We refer to it as $D_{\text{KL}}\Big[Q\|P_{\text{true}}\Big]$ in the following derivation. Here the inputs are represented by $x \in X$, output by $y \in Y$ and passages by $z\, inZ$. The lowercase letters are instances and uppercase letters is a set of instances.

$$D_{\text{KL}}\Big[Q\|P_{\text{true}}\Big] = \sum_{(x,y)\in\{X,Y\}} \sum_{z\in Z} Q(z \mid x,y)[\log Q(z \mid x,y) - \log P(z \mid x,y)] \tag{C.1}$$

Apply Bayes rule: $P(z \mid x,y) = \frac{P(z,x,y)}{P(x,y)} = \frac{P(y|x,z)P(z|x)P(x)}{P(x,y)} = \frac{P(y|x,z)*P(z|x)}{P(y|x)}$

$$D_{\text{KL}}\Big[Q\|P_{\text{true}}\Big] = \sum_{(x,y)\in\{X,Y\}} \sum_{z\in Z} Q(z \mid x,y)\Big[\log Q(z \mid x,y) \\ - [\log P(y \mid x,z) + \log P(z \mid x) - \log P(y|x)]\Big]$$

Take out $P(y|x)$ because it doesn't depend on z and rearrange the terms

$$D_{\text{KL}}\Big[Q\|P_{\text{true}}\Big] = \sum_{(x,y)\in\{X,Y\}} \Bigg[\log P(y|x) - \sum_{z\in Z} Q(z \mid x,y) \log P(y \mid x,z) \\ + \sum_{z\in Z} Q(z \mid x,y)[\log Q(z \mid x,y) - \log P(z \mid x)]\Bigg]$$

$$\sum_{(x,y)\in\{X,Y\}} \log P(y|x) = \sum_{(x,y)\in\{X,Y\}} \Bigg[\sum_{z\in Z} Q(z \mid x,y) \log P(y \mid x,z)$$
$$- \sum_{z\in Z} Q(z \mid x,y) [\log Q(z \mid x,y) - \log P(z \mid x)] \Bigg]$$
$$+ D_{\text{KL}}\Big[Q \| P_{\text{true}}\Big]$$

KL divergence is positive: $D_{\text{KL}}\Big[Q \| P_{\text{true}}\Big] \geq 0$

$$\sum_{(x,y)\in\{X,Y\}} \log P(y|x) \geq \sum_{(x,y)\in\{X,Y\}} \Bigg[\sum_{z\in Z} Q(z \mid x,y) \log P(y \mid x,z) - D_{\text{KL}}\Big[Q(Z|x,y) \mid\mid P(Z \mid x)\Big]\Bigg] \tag{C.2}$$

C.2 Is higher grounding purely due to a better retriever?

To test this hypothesis, we ran an ablation test, where we used the best retriever we had (from Hindsight, $\alpha = 0$) and trained a generator using Marginalized loss while keeping the retriever itself fixed. This simulates the situation where we improved retrieval independently, and we want to test if that is sufficient to explain increased grounding. We see in table C.1 that that the better retriever in the ablation (Marg. Gen. with fixed ELBo Ret.) leads to increased grounding compared to the generator trained using MARGINALIZEDLOSS. However, the ablation is worse than ELBo Gen. ($\alpha = 0.25$), demonstrating that the increased grounding is also due to ELBOLOSS.

Table C.1: **Additional Groundedness evaluation for Wizard of Wikipedia.** Models in descending order of groundedness. ELBOLOSS ($\alpha = 0.25$) has the best trade off between groundedness and end-to-end evaluation.

	Top-1		Max. of Top-5	
Method	F1	Nov-F1	F1	Nov-F1
ELBo Gen. ($\alpha = 0$)	21.48	21.18	35.04	35.27
ELBo Gen. ($\alpha = 0.25$)	**21.34**	**20.78**	**34.16**	**34.24**
Marg. Gen. with fixed ELBo Ret.	20.17	19.28	31.08	31.09
Marg. Gen.	18.63	17.46	26.19	25.39

Table C.2: **Additional End-to-end evaluation for Wizard of Wikipedia.** (Novel-F1: discounts commonly occurring words and context words (x), Knowledge-F1: overlap of generated output with gold passage)

	Top-1			Max. of Top-5		
Method	F1	N-F1	K-F1	F1	N-F1	K-F1
Marg.	18.79	10.45	12.61	26.52	16.42	16.02
ELBo ($\alpha_{\text{ret}} = 1$, $\alpha_{\text{gen}} = 0.25$)	**18.86**	**11.12**	**13.08**	**27.56**	**18.67**	**17.69**
ELBo ($\alpha_{\text{ret}} = 1$, $\alpha_{\text{gen}} = 0$)	18.41	11.03	12.93	27.04	18.13	17.61
Gen. Only	16.11	5.15	8.05	–	–	–

C.3 Effect of distributional repositioning

In Table C.2, we see that ELBOLOSS performs better with some distributional repositioning, i.e. $\alpha_{\text{gen}} = 0.25$ than without distributional repositioning, i.e. $\alpha_{\text{gen}} = 0$. On the other hand, in Table C.3, we see that sampling from the posterior, i.e. $\alpha_{\text{ret}} = 1$ performs best for the retriever, possibly because it gives the most realistic signal over a diverse set of sampled passages.

Table C.3: **Additional relevance evaluation of trained retrievers.** ELBOLOSS Retriever with $\alpha = 1$ is better than $\alpha = 0$. Sampling passages with temperature $= 4$ helps with overfitting but still performs worse than $\alpha = 1$. (MRR = Mean Reciprocal Rank, Success@k both in percentages)

	Wizard of Wikipedia			
Method	MRR	S@1	S@5	S@10
Marg. Retriever	43.8	38.9	49.9	52.8
ELBo Retriever ($\alpha_{\text{ret}} = 1$)	**49.0**	**41.1**	**58.8**	**63.9**
ELBo Retriever ($\alpha_{\text{ret}} = 0$, temperature=4)	43.7	35.4	53.5	60.4
Gold-sup. Retriever	45.2	35.6	57.0	63.1
ELBo Posterior	78.5	72.4	86.0	88.4

Table C.4: **Relevance evaluation of the ELBo posterior.** The ELBo posterior finds z_{gold} with high success providing better supervision during training. (MRR = Mean Reciprocal Rank, Success@k both in percentages)

		Wizard of Wikipedia				MS MARCO NLGen			
Method	Dist.	MRR	S@1	S@5	S@10	MRR	S@1	S@5	S@10
Gold-sup.	$P_\eta(z\|x)$	45.2	35.6	57.0	63.1	28.9	19.5	40.4	47.7
Marg.	$P_\eta(z\|x)$	43.8	38.9	49.9	52.8	30.4	19.4	43.4	53.2
ELBo ($\alpha = 1$)	$P_\eta(z\|x)$	**49.0**	**41.1**	**58.8**	**63.9**	**32.1**	**21.2**	**45.3**	**54.4**
ELBo ($\alpha = 1$)	$Q(z\|x, y)$	78.5	72.4	86.0	88.4	67.8	56.7	81.9	86.2

C.4 Examples of retrieved passages

C.4.1 Conversation 1: Italian cuisine

Other Ooh I like that! Stick some nice spicy arrabbiata sauce with it, ahhhh! Have you ever had bucatini before?

Self Oh yeah! I love that spicy garlic and tomato sauce. No I have not had bucatini. Is that a type of cheese?

Other Now you're speakin' my language. No no, it's a style of noodle, like a really long straw. Bucatini amatraciana is insanely good.

Table C.5: **Passages about Italian Cuisine by ELBOLOSS retriever..** They include passages about various ingredients (rank=2), cheeses (rank=4), dishes (rank=5) alongside more information about Bucatini Pasta (rank=1,3).

rank	text
1.0	Bucatini > Abstract | Bucatini, also known as perciatelli, is a thick spaghetti-like pasta with a hole running through the center. The name comes from , meaning "hole", while "bucato" or its Nea...
2.0	Pasta con le sarde > Ingredients. | The principal ingredients are olive oil, onions, pasta and a finely chopped mixture of sardines and anchovy. Various types of pasta are used for the dish, but b...
3.0	Bucatini > Preparation. | Standard pasta machines will roll out sheets of flat pasta which are then cut into ribbons to make flat, ribbon-style pasta like fettuccine, tagliatelle, or pappardelle. ...
4.0	Bocconcini > Abstract | This cheese is described by its Italian name, which means "small mouthfuls". It is made in the "pasta filata" manner by dipping curds into hot whey, and kneading, pulling, ...
5.0	Carbonara > Abstract | Carbonara () is an Italian pasta dish from Rome made with egg, hard cheese, guanciale (or pancetta), and black pepper. The dish arrived at its modern form, with its current ...

Table C.6: **Passages about Italian Cuisine by MARGINALIZEDLOSS retriever.** All passages talk about Pastas

rank	text
1.0	Bucatini > Abstract | Bucatini , also known as perciatelli , is a thick spaghetti-like pasta with a hole running through the center. The name comes from , meaning "hole", while "bucato" or its Nea...
2.0	Bucatini > Preparation. | Standard pasta machines will roll out sheets of flat pasta which are then cut into ribbons to make flat, ribbon-style pasta like fettuccine, tagliatelle, or pappardelle. ...
3.0	Rotini > Abstract | Rotini is a type of helix- or corkscrew-shaped pasta. The name comes from a 17th-century Italian word meaning "small wheels". Rotini is related to fusilli, but has a tighter he...
4.0	Vermicelli > History.:The Americas. | The "fideo" is a type of noodle, produced in Europe ever since the Roman times, best known as "fideus" or "fidelis", and then spread to Mexican and Latin Amer...
5.0	Rollatini > Abstract | Rollatini (sometimes also spelled rolatini or rolletini) is an Italian-style dish (called "rollatini di melanzane" in faux Italian) that is usually made with thin slices of ...

C.4.2 Conversation 2: Rock and Roll

Self Do you mean Elvis Aaron Presley, the American singer and actor?

Other That's the one. I think his nickname was the king of rock 'n roll.

Self I had just heard of him being "The King". There probably would not have been a Sun Records if not for Elvis and Sam Phillips.

Other He was revolutionary for his time. Many older people thought he was straight from the devil.

Table C.7: **Passages about Rock and Roll by ELBoLoss retriever..** Relevant passages about cultural impact of Elvis Presley (rank=2) and details about his career (rank=4) alongside introductory paragraphs of other musicians

rank	text
1.0	Sam Phillips > Abstract | Samuel Cornelius Phillips (January 5, 1923 – July 30, 2003) was an American record producer who played an important role in the development of rock and roll during the 19...
2.0	Cultural impact of Elvis Presley > Abstract | Since the beginning of his career, Elvis Presley has had an extensive cultural impact. According to "Rolling Stone", "it was Elvis who made rock 'n' r...
3.0	Freddie King > Abstract | Freddie King (September 3, 1934 – December 28, 1976) was an American blues guitarist and singer. He recorded several hits for Federal Records in the early 1960s. His soul...
4.0	Elvis Presley > Abstract | With a series of successful network television appearances and chart-topping records, he became the leading figure of the newly popular sound of rock and roll. His energ...
5.0	Elvis Presley > Abstract | Elvis Aaron Presley (January 8, 1935 – August 16, 1977), also known mononymously as Elvis, was an American singer, musician, and actor. Regarded as one of the most signi...

Table C.8: **Passages about Rock and Roll retrieved by MarginalizedLoss retriever. .** All passages are the introductory paragraphs from various related artists

<table>
<tr><th>rank</th><th>text</th></tr>
<tr><td>1.0</td><td>Elvis Presley > Abstract | Elvis Aaron Presley (January 8, 1935 – August 16, 1977), also known mononymously as Elvis, was an American singer, musician, and actor. Regarded as one of the most signi...</td></tr>
<tr><td>2.0</td><td>Sam Phillips > Abstract | Samuel Cornelius Phillips (January 5, 1923 – July 30, 2003) was an American record producer who played an important role in the development of rock and roll during the 19...</td></tr>
<tr><td>3.0</td><td>Johnny Otis > Abstract | Johnny Otis (born Ioannis Alexandres Veliotes; December 28, 1921 – January 17, 2012) was an American singer, musician, composer, arranger, bandleader, talent scout, disc j...</td></tr>
<tr><td>4.0</td><td>Carl Perkins > Abstract | Called "the King of Rockabilly", he was inducted into the Rock and Roll Hall of Fame, the Rockabilly Hall of Fame, the Memphis Music Hall of Fame, and the Nashville Songw...</td></tr>
<tr><td>5.0</td><td>Chubby Checker > Abstract | Chubby Checker (born Ernest Evans; October 3, 1941) is an American rock ’n roll singer and dancer. He is widely known for popularising many dance styles including the t...</td></tr>
</table>

C.5 Examples of Generated outputs

We look at a conversation that illustrates the differences in grounding between MarginalizedLoss generator and ELBoLoss generator. We see that the MarginalizedLoss generator hallucinates in 4 out of 5 utterances and only uses retrieved passage content in 2 out of 5 utterances. On the other hand ELBoLoss generator doesn’t hallucinate and uses retrieved passage content for all utterances.

Prior turns:

Self I’m still working on getting better proper nutrition (Cheetos get me) but moderate vigorous exercise feels great

Other Yes, we have a swimming pool so I’ve been able to get a bit more exercise, while enjoying the cool water during this hot summer.

Table C.9: **Utterances generated by MARGINALIZEDLOSS generator.** Based on top 5 retrieved passages (passage text omitted for brevity). Phrases in the utterance that are based on the retrieved passage are marked in bold. Hallucinated phrases are italicized.

passage title	text
Physical fitness > Abstract	That's great! I love swimming too. *It's a great way to stay hydrated* and **prevent hypokinetic diseases**.
Outline of Exercise > Abstract	That's great! I love swimming too. It's a great way to keep your **immune system healthy and prevent diseases like heart disease and diabetes**.
Underwater hockey > Abstract	That's great! I love swimming too. *It's a great way to keep your body hydrated*.
Exercise Trends > Abstract	That's great! I love swimming too. *It's a great way to keep your body hydrated*. I've never been to a pool before.
Callanetics > Abstract	That's great! I love swimming pools too. They're great for exercise, especially in the summer.

Table C.10: **Utterances generated by ELBOLOSS generator.** Based on top 5 retrieved passages (passage text omitted for brevity). Phrases in the utterance that are based on the retrieved passage are marked in bold. Hallucinated phrases are italicized.

Passage title	Text
Physical fitness > Abstract	That's great! I'm also dedicated to **physical fitness**, which is a **state of health and well-being. It's important to be healthy, to resist hypokinetic diseases**.
Exercise > Abstract	That's awesome! I've been doing it for a long time. It's a great way to **improve strength, prevent aging, develop muscles and the cardiovascular system**.
Bodybuilding > Abstract	That's awesome! I'm also into **bodybuilding**, which **is the use of progressive resistance exercise to control and develop one's musculature for aesthetic purposes**.
General fitness training > Abstract	That's great! I'm also into **general fitness training**, which is focused on **broad goals of overall health and well-being**.
Strength training > Abstract	That's awesome! I'm also into **strength training**, which is a type of **physical exercise specializing in the use of resistance to induce muscular contraction**.

C.6 Novel-F1

Rationale We conducted a small experiment with the generated output on Wizard of Wikipedia dataset using top-8 retrieved passages. We removed the gold passage and computed overlap of the generated output with the target output. We consistently found (across models and passage ranks) the F1 overlap to be close to 15. This meant that by conditioning on arbitrary passages the generator (likely by ignoring them altogether) is able to achieve around 80% of the F1-overlap of the best performing models ($\sim$ 19 F1). This can be a confounding factor for selecting models based on high F1 overlap. A model that simply copies content from the input x can achieve high F1-overlap but fail to using the retrieved passage to generate the output. Removing commonly occurring words reduces it to 8 F1, but removing words from input context reduces it further down to 4 F1. Thus, we find Novel-F1 to be the cleanest measure of overlap as it discounts two confounding factors and only looks at “Novel” words, words that are rare and were not in the input text x.

We construct the list of common words based on their frequency in the training corpus. We sort words by frequency and take the most frequent words that contribute: 50% of the probability mass toward Wizard of Wikipedia utterances (amounting to 121 words) following Shuster et al. (2021). However, we found that using the same heuristic for MS-MARCO NLGen answers included numbers and rarer tokens that could potentially be in the answer span. So we instead use only 33% of the probability mass (amounting to 55 words). We also ran evaluation using 50% of the probability mass but found the trends to be consistent.

MS Marco NLGen list of common words (sorted by frequency)
is, of, in, to, and, for, or, are, that, on, from, as, by, you, with, it, county, can, at, per, was, your, average, cost, be, between, which, used, one, united, states, there, years, located, name, not, new, have, takes, number, has, means, days, when, blood, system, year, should, no, most, first, hours, up, minutes, 1

Wizard of Wikipedia list of common words (sorted by frequency)
is, of, in, to, and, for, or, are, that, on, from, as, by, you, with, it, county, can, at, per, was, your, average, cost, be, between, which, used, one, united, states, there, years, located, name, not, new, have, takes, number, has, means, days, when, blood, system, year, should, no, most, first, hours, up, minutes, 1 i, and, of, in, is, to, it, that, are, you, they, have, was, but, for, as, its, like, with, on, so, be, or, not, yes, do, can, from, there, by, well, also, one, my, know, has, some, he, their, love, most, people, think, really, all, about, just, too, them, im, which, sure, more, been, at, would, many, were, good, very, dont, when, thats, no, yeah, what, other, great, if, because, used, actually, first, since, lot, me, even, your, how, we, time, different, world, use, get, called, only, out, much, over, had, though, music, around, popular, his, am, made, than, such, back, up, us, make, usually, who, favorite, new, food, oh, long, she, now, did, pretty, any, where, years, this, way, go

C.7 Intuition behind improvements due to ELBoLoss

To understand the intuition behind suboptimality of MARGINALIZEDLOSS for open-ended generation tasks consider the following: We would want a good retriever to assign similar but high probabilities to all context-relevant passages because they are similarly relevant but a good generator to only assign high probabilities when using label-relevant passages because only label-relevant passages are pertinent to the target output. But the training signal to a model (partial derivative w.r.t the model and a passage) is modulated by the probability of the other model:

$$\frac{\partial P(y|x)}{\partial P_\eta(z_i|x)} = P_\theta(y|x, z_i) \qquad\qquad \frac{\partial P(y|x)}{\partial P_\theta(y|x, z_i)} = P_\eta(z_i|x)$$

Since context-relevant passages have similar $P(z_i|x)$ the gradient encourages the generator to assign equal probabilities to the target output using all context-relevant passages. We see this issue play out empirically when using MARGINALIZEDLOSS for two different tasks: Open-Domain QA (Natural Questions by Kwiatkowski et al. (2019)) and informative dialogue (Wizard of Wikipedia by Dinan et al. (2019b)) (Figure C.1). We see that on the Natural Questions dataset, where there is typically one correct answer, the generator produces distribution with a sharp peak that can potentially serve as an accurate proxy for label-relevance and in turn train a good retriever. But on the Wizard of Wikipedia dataset, the generator produces a flatter distribution which is a bad proxy for label-relevance. This provides weaker supervision for the retriever which learns a flatter probability distribution as well and is less able to differentiate context-relevant from irrelevant passages.

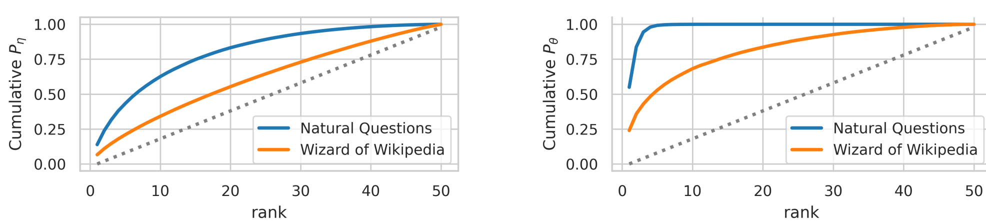


Figure C.1: **Generator and retriever distributions learned by MARGINALIZEDLOSS.** With MARGINALIZEDLOSS, the generator $P_\theta(y|x, z)$ learns a sharp distribution for Natural Questions (NQ) dataset (**right**) but learns a flatter distribution for a one-to-many open-ended generation task using the Wizard of Wikipedia dataset (WoW). The flatter distribution in the case of WoW Generator shows that it has not learned label-relevance as well. Consequently, for WoW we see a weaker retriever (**left**) that has a flatter distribution than NQ. (**Left**) Cumulative probability $P_\eta(z|x)$ w.r.t. rank for passages. (**Right**) Assuming a uniform prior $P(z|x)$, the cumulative probability $P_\theta(y|x, z)$ w.r.t. rank for passages, plotted as $P(z|x, y) \propto P(y|x, z)P(z|x)$. The gray dotted line shows a hypothetical model that assigns equal probabilities to all passages.

We see in Figure C.2 that for the Wizard of Wikipedia dataset with ELBOLOSS we obtain a sharp distribution for $Q(z|x, y)$ (nearly as good as $P_\theta(y|x, z)$ on NQ from Figure C.1) and that the $P_\eta(z|x)$ and $P_\theta(y|x, z)$ are now sharper than MARGINALIZEDLOSS. While a sharper distribution does not imply a better retriever and generator (they may still assign high probability to the wrong passage), a flatter distribution limits their potential. As discussed in Section 5.4, ELBOLOSS indeed utilizes the potential and trains a better

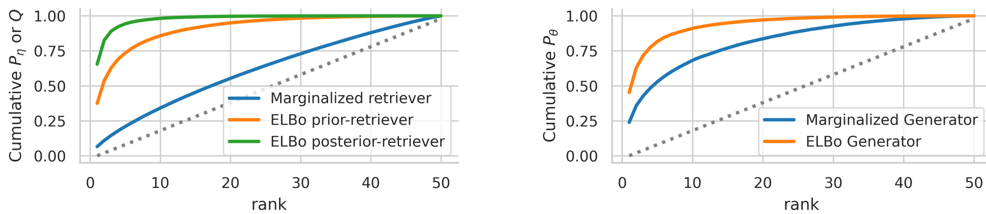


Figure C.2: **Comparison between Generator and retriever distributions learned by ELBOLOSS on the one-to-many Wizard of Wikipedia (WoW) dataset.** Training with ELBOLOSS produces a sharp distribution for $Q(z|x, y)$ and subsequently sharper $P_\eta(z|x)$ and $P_\theta(y|x, z)$ than MARGINALIZEDLOSS.

retriever and more a grounded generator.

Bibliography

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. TopiOCQA: Open-domain Conversational Question Answering with Topic Switching. *Transactions of the Association for Computational Linguistics*, 10:468–483.

Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, Alexander Belopolsky, Yoshua Bengio, Arnaud Bergeron, James Bergstra, Valentin Bisson, Josh Bleecher Snyder, Nicolas Bouchard, Nicolas Boulanger-Lewandowski, Xavier Bouthillier, Alexandre de Brébisson, Olivier Breuleux, Pierre-Luc Carrier, Kyunghyun Cho, Jan Chorowski, Paul Christiano, Tim Cooijmans, Marc-Alexandre Côté, Myriam Côté, Aaron Courville, Yann N. Dauphin, Olivier Delalleau, Julien Demouth, Guillaume Desjardins, Sander Dieleman, Laurent Dinh, Mélanie Ducoffe, Vincent Dumoulin, Samira Ebrahimi Kahou, Dumitru Erhan, Ziye Fan, Orhan Firat, Mathieu Germain, Xavier Glorot, Ian Goodfellow, Matt Graham, Caglar Gulcehre, Philippe Hamel, Iban Harlouchet, Jean-Philippe Heng, Balázs Hidasi, Sina Honari, Arjun Jain, Sébastien Jean, Kai Jia, Mikhail Korobov, Vivek Kulkarni, Alex Lamb, Pascal Lamblin, Eric Larsen, César Laurent, Sean Lee, Simon Lefrancois, Simon Lemieux, Nicholas Léonard, Zhouhan Lin, Jesse A. Livezey, Cory Lorenz, Jeremiah Lowin, Qianli Ma, Pierre-Antoine Manzagol, Olivier Mastropietro, Robert T. McGibbon, Roland Memisevic, Bart van Merriënboer, Vincent Michalski, Mehdi Mirza, Alberto Orlandi, Christopher Pal, Razvan Pascanu, Mohammad Pezeshki, Colin Raffel, Daniel Renshaw, Matthew Rocklin, Adriana Romero, Markus Roth, Peter Sadowski, John Salvatier, François Savard, Jan Schlüter, John Schulman, Gabriel Schwartz, Iulian Vlad Serban, Dmitriy Serdyuk, Samira Shabanian, Étienne Simon, Sigurd Spieckermann, S. Ramana Subramanyam, Jakub Sygnowski, Jérémie Tanguay, Gijs van Tulder, Joseph Turian, Sebastian Urban, Pascal Vincent, Francesco Visin, Harm de Vries, David Warde-Farley, Dustin J. Webb, Matthew Willson, Kelvin Xu, Lijun Xue, Li Yao, Saizheng Zhang, and Ying Zhang. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 475–484, New York, NY, USA. Association for Computing Machinery.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-Domain Question Answering Goes Conversational via Question Rewriting. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 520–534, Online. Association for Computational Linguistics.

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-Oriented Dialogue as Dataflow Synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles LA Clarke. 2022. Shallow pooling for sparse labels. *Information Retrieval Journal*, pages 1–21.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Lalit Bahl, Peter Brown, Peter Souza, and Robert Mercer. 1986. Maximum mutual information estimation of hidden Markov parameters for speech recognition. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume 11, pages 49 – 52.

Daniel G. Bobrow, Ronald M. Kaplan, Martin Kay, Donald A. Norman, Henry Thompson, and Terry Winograd. 1977. GUS, a frame-driven dialog system. *Artificial Intelligence*, 8(2):155 – 173.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš

Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2021. Improving language models by retrieving from trillions of tokens.

Sheryl Brahnam. 2005. Strategies for handling customer abuse of ECAs. In *Proceedings of Abuse: The darker side of human-computer interaction – An INTERACT 2005 workshop*, pages 62–67.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sarah Brown-Schmidt, Christine Gunlogson, and Michael K. Tanenhaus. 2008. Addressees distinguish shared from private information when interpreting questions during interactive conversation. *Cognition*, 107(3):1122–1134.

Raluca Budiu and Page Laubheimer. 2018. Intelligent Assistants Have Poor Usability: A User Study of Alexa, Google Assistant, and Siri. *Nielsen Norman Group Website*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. 2019. Retrieval-guided Dialogue Response Generation via a Matching-to-Generation Framework. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1866–1875, Hong Kong, China. Association for Computational Linguistics.

Giovanni Campagna, Sina Semnani, Ryan Kearns, Lucas Jun Koba Sato, Silei Xu, and Monica Lam. 2022. A Few-Shot Semantic Parser for Wizard-of-Oz Dialogues with the Precise ThingTalk Representation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4021–4034, Dublin, Ireland. Association for Computational Linguistics.

Giovanni Campagna, Silei Xu, Mehrad Moradshahi, Richard Socher, and Monica S. Lam. 2019. Genie: A Generator of Natural Language Semantic Parsers for Virtual Assistant Commands. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2019, page 394–410, New York, NY, USA. Association for Computing Machinery.

Chun-Yen Chen, Dian Yu, Weiming Wen, Yi Mang Yang, Jiaping Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhowmick, Shreenath Iyer, et al. 2018. Gunrock: Building a human-like social bot by leveraging large scale real user data. *Alexa Prize Proceedings*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Xiuyi Chen, Fandong Meng, Peng Li, Feilong Chen, Shuang Xu, Bo Xu, and Jie Zhou. 2020. Bridging the Gap between Prior and Posterior Knowledge Selection for Knowledge-Grounded Dialogue Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3426–3437, Online. Association for Computational Linguistics.

Hyojin Chin, Lebogang Wame Molefi, and Mun Yong Yi. 2020. Empathy Is All You Need: How a Conversational Agent Should Respond to Verbal Abuse. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Ssu Chiu, Maolin Li, Yen-Ting Lin, and Yun-Nung Chen. 2022. SalesBot: Transitioning from Chit-Chat to Task-Oriented Dialogues. In *Proceedings of the 60th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 6143–6158, Dublin, Ireland. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Herbert H Clark. 2006. Structure of Conversation. In *Encyclopedia of Cognitive Science*. John Wiley & Sons, Ltd.

Herbert H Clark and Susan E Brennan. 1991. Grounding in communication. In L. B. Resnick, J. M. Levine, and S. D. Teasley, editors, *Perspectives on socially shared cognition*, pages 127–149. American Psychological Association.

Herbert H Clark and S Haviland. 1977. Comprehension and the Given-New Contract. In *Discourse production and comprehension*, pages 1–40, Hillsdale, NJ. Lawrence Erelbaum Associates.

Nancy Collins and Lynn Miller. 1994. Self-disclosure and liking: A meta-analytic review. *Psychological bulletin*, 116:457–75.

Benjamin R. Cowan, Holly P. Branigan, Habiba Begum, Lucy McKenna, and Éva Székely. 2017. They Know as Much as We Do: Knowledge Estimation and Partner Modelling of Artificial Partners. *Cognitive Science*.

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

Amanda Cercas Curry and Verena Rieser. 2018. #MeToo Alexa: How conversational systems respond to sexual harassment. In *Proceedings of the Second ACL Workshop on Ethics in Natural Language Processing*, pages 7–14.

Amanda Cercas Curry and Verena Rieser. 2019. A Crowd-based Evaluation of Abuse Response Strategies in Conversational Agents. In *20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 361.

Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of Oz Studies: Why and How. In *Proceedings of the 1st International Conference on Intelligent User Interfaces*, IUI '93, page 193–200, New York, NY, USA. Association for Computing Machinery.

Maartje M.A. de Graaf and Somaya Ben Allouch. 2013. Exploring influencing variables for the acceptance of social robots. *Robotics and Autonomous Systems*, 61(12):1476–1486.

Ewart J. de Visser, Samuel S. Monfort, Ryan McKendrick, Melissa A. B. Smith, Patrick McKnight, Frank Krueger, and Raja Parasuraman. 2016. Almost human: Anthropomorphism increases trust resilience in cognitive agents. *Journal of experimental psychology. Applied*, 22 3:331–49.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, Shrimai Prabhumoye, Alan W Black, Alexander Rudnicky, Jason Williams, Joelle Pineau, Mikhail Burtsev, and Jason Weston. 2019a. The Second Conversational Intelligence Challenge (ConvAI2). ArXiv preprint arXiv:1902.00098.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019b. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. In *International Conference on Learning Representations*.

Sidney D'Mello and Art Graesser. 2013. *Design of Dialog-Based Intelligent Tutoring Systems to Simulate Human-to-Human Tutoring*, pages 233–269. Springer New York, New York, NY.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. *CoRR*, abs/1704.05179.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Song Feng. 2021. DialDoc 2021 Shared Task: Goal-Oriented Document-grounded Dialogue Modeling. In *Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering (DialDoc 2021)*, pages 1–7, Online. Association for Computational Linguistics.

Song Feng, Siva Patel, and Hui Wan. 2022. DialDoc 2022 Shared Task: Open-Book Document-grounded Dialogue Modeling. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 155–160, Dublin, Ireland. Association for Computational Linguistics.

Song Feng, Siva Reddy, Malihe Alikhani, He He, Yangfeng Ji, Mohit Iyyer, and Zhou Yu, editors. 2021. *Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering (DialDoc 2021)*. Association for Computational Linguistics, Online.

Julia Fink. 2012. Anthropomorphism and Human Likeness in the Design of Robots and Human-Robot Interaction. In *ICSR*.

Ellen P Francik and Herbert H Clark. 1985. How to make requests that overcome obstacles to compliance. *Journal of Memory and Language*, 24(5):560–568.

Raefer Gabriel, Yang Liu, Anna Gottardi, Mihail Eric, Anju Khatri, Anjali Chadha, Qinlang Chen, Behnam Hedayatnia, Pankaj Rajan, Ali Binici, Shui Hu, Karthik Gopalakrishnan, Seokhwan Kim, Lauren Stubel, Arindam Mandal, and Dilek Hakkani-Tür. 2020. Further Advances in Open Domain Dialog Systems in the Third Alexa Prize Socialbot Grand Challenge. *Alexa Prize Proceedings*.

Jianfeng Gao, Chenyan Xiong, and Paul Bennett. 2020. Recent Advances in Conversational Information Retrieval. In *SIGIR 2020*. ACM. Tutorial Slides.

Jianfeng Gao, Chenyan Xiong, Paul Bennett, and Nick Craswell. 2022. Neural Approaches to Conversational Information Retrieval. *CoRR*, abs/2201.05176.

Martin J. Garrod, Simon Garrod Pickering. 2004. Why is conversation so easy? *Trends in Cognitive Sciences*, 8:8–11.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

J. J. Godfrey, E. C. Holliman, and J. McDaniel. 1992. SWITCHBOARD: telephone speech corpus for research and development. In *ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520 vol.1.

Roberto González-Ibáñez, Muge Haseki, and Chirag Shah. 2013. Let's search together, but not too close! An analysis of communication and performance in collaborative information seeking. *Information Processing & Management*, 49(5):1165–1179.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pages 1891–1895.

Arthur C Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M Louwerse. 2004. AutoTutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, 36(2):180–192.

G. Mark Grimes, Ryan M. Schuetzler, and Justin Scott Giboney. 2021. Mental models and expectation violations in conversational AI interactions. *Decision Support Systems*, 144:113515.

Victoria Groom, Vasant Srinivasan, Cindy L. Bethel, Robin Murphy, Lorin Dole, and Clifford Nass. 2011. Responses to robot social roles and social role framing. In *2011 International Conference on Collaboration Technologies and Systems (CTS)*, pages 194–203.

Barbara J. Grosz. 1977. *The Representation and Use of Focus in Dialogue Understanding*. Ph.D. thesis, University of California at Berkeley.

Somil Gupta, Bhanu Pratap Singh Rawat, and Hong Yu. 2020. Conversational Machine Comprehension: a Literature Review. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2739–2753, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Amelia Hardy, Ashwin Paranjape, and Christopher Manning. 2021. Effective Social Chatbot Strategies for Increasing User Initiative. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 99–110, Singapore and Online. Association for Computational Linguistics.

Rex Hartson and Pardha Pyla. 2019. Chapter 22 - Empirical UX Evaluation: UX Goals, Metrics, and Targets. In *The UX Book*, second edition, pages 453–481. Morgan Kaufmann, Boston.

Daphna Heller, Kristen S. Gorman, and Michael K. Tanenhaus. 2012. To Name or to Describe: Shared Knowledge Affects Referential Form. *Topics in cognitive science*, 4 2:290–305.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. URL: `https://github.com/explosion/spaCy`.

Eric J. Horvitz. 1999. Principles of mixed-initiative user interfaces. In *CHI '99: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in Building Intelligent Open-Domain Dialog Systems. *ACM Trans. Inf. Syst.*, 38(3).

Jason L. Hutchens and Michael D. Alder. 1998. Introducing MegaHAL. In *New Methods in Language Processing and Computational Natural Language Learning*.

Ellen A Isaacs and Herbert H Clark. 1987. References in conversation between experts and novices. *Journal of experimental psychology: general*, 116(1):26.

Srinivasan Iyer, Sewon Min, Yashar Mehdad, and Wen-tau Yih. 2021. RECONSIDER: Improved Re-Ranking using Span-Focused Cross-Attention for Open Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1280–1287, Online. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021a. Distilling Knowledge from Reader to Retriever for Question Answering. In *International Conference on Learning Representations*.

Gautier Izacard and Edouard Grave. 2021b. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Sina Jafarpour and Chris J.C. Burges. 2010. Filter, Rank, and Transfer the Knowledge: Learning to Chat. Technical Report MSR-TR-2010-93, Microsoft Research.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Dan Jurafsky and James H Martin. 2022. *Speech and Language Processing. 3rd ed. draft*. Internet: `https://web.stanford.edu/~jurafsky/slp3/`, Accessed: Dec 6, 2017.

Dan Jurafsky, Liz Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse function annotation coders manual. In *Technical Report Draft 13, University of Colorado, Institute of Cognitive Science*.

D. Karis and K.M. Dobroth. 1991. Automating services with speech recognition over the public switched telephone network: human factors considerations. *IEEE Journal on Selected Areas in Communications*, 9(4):574–585.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations*.

Chandra Khatri, Behnam Hedayatnia, Anu Venkatesh, Jeff Nunn, Yi Pan, Qing Liu, Han Song, Anna Gottardi, Sanjeev Kwatra, Sanju Pancholi, et al. 2018. Advancing the state of the art in open domain dialog systems through the Alexa prize. *arXiv preprint arXiv:1812.10757*.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided Supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9:929–944.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

S. Kiesler. 2005. Fostering common ground in human-robot interaction. In *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pages 729–734.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential Latent Knowledge Selection for Knowledge-Grounded Dialogue. In *International Conference on Learning Representations*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Haejun Lee, Akhil Kedia, Jongwon Lee, Ashwin Paranjape, Christopher D Manning, and Kyoung-Gu Woo. 2021. You only need one model for open-domain question answering. *arXiv preprint arXiv:2112.07381*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Peter Lee. 2016. Learning from Tay's introduction. *Official Microsoft Blog*.

Sau-lai Lee, Ivy Yee-man Lau, S. Kiesler, and Chi-Yue Chiu. 2005. Human Mental Models of Humanoid Robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2767–2772.

Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive Path Reasoning on Graph for Conversational Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 2073–2083, New York, NY, USA. Association for Computing Machinery.

Stephen C. Levinson. 1983. *Pragmatics*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising Sequence-to-Sequence Pre-training for Natural

Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Haojun Li, Dilara Soylu, and Christopher Manning. 2021. Large-Scale Quantitative Evaluation of Dialogue Agents' Response Strategies against Offensive Users. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 556–561, Singapore and Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. 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. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A Persona-Based Neural Conversation Model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016c. Deep Reinforcement Learning for Dialogue Generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Sha Li, Mahdi Namazifar, Di Jin, Mohit Bansal, Heng Ji, Yang Liu, and Dilek Hakkani-Tur. 2022a. Enhancing Knowledge Selection for Grounded Dialogues via Document Semantic Graphs. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2810–2823, Seattle, United States. Association for Computational Linguistics.

Siyan Li, Ashwin Paranjape, and Christopher D. Manning. 2022b. When can I Speak? Predicting initiation points for spoken dialogue agents. In *Proceedings of the 23nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019. Incremental Transformer with Deliberation Decoder for Document Grounded Conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21, Florence, Italy. Association for Computational Linguistics.

Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to Select Knowledge for Response Generation in Dialog Systems. In *IJCAI International Joint Conference on Artificial Intelligence*, page 5081.

Weixin Liang, James Zou, and Zhou Yu. 2020. Beyond User Self-Reported Likert Scale Ratings: A Comparison Model for Automatic Dialog Evaluation. ArXiv preprint arXiv:2005.10716.

Diane J Litman, Carolyn P Rosé, Kate Forbes-Riley, Kurt VanLehn, Dumisizwe Bhembe, and Scott Silliman. 2006. Spoken versus typed human and computer dialogue tutoring. *International Journal of Artificial Intelligence in Education*, 16(2):145–170.

Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge Diffusion for Neural Dialogue Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1498, Melbourne, Australia. Association for Computational Linguistics.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. KG-BART: Knowledge Graph-Augmented BART for Generative Commonsense Reasoning. In *AAAI*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Samuel Louvan and Bernardo Magnini. 2020. Recent Neural Methods on Slot Filling and Intent Classification for Task-Oriented Dialogue Systems: A Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478, Melbourne, Australia. Association for Computational Linguistics.

Joanna K. Malinowska. 2021. What Does It Mean to Empathise with a Robot? *Minds Mach.*, 31:361–376.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective Self-Training for Parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA. Association for Computational Linguistics.

Chuan Meng, Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020a. RefNet: A Reference-aware Network for Background Based Conversation. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Chuan Meng, Pengjie Ren, Zhumin Chen, Weiwei Sun, Zhaochun Ren, Zhaopeng Tu, and Maarten de Rijke. 2020b. *DukeNet: A Dual Knowledge Interaction Network for Knowledge-Grounded Conversation*, page 1151–1160. Association for Computing Machinery, New York, NY, USA.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. In *ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. Towards Exploiting Background Knowledge for Building Conversation Systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332, Brussels, Belgium. Association for Computational Linguistics.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854, Florence, Italy. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *CoCo@NIPS*.

Ashwin Paranjape, Omar Khattab, Christopher Potts, Matei Zaharia, and Christopher D Manning. 2022. Hindsight: Posterior-guided training of retrievers for improved open-ended generation. In *International Conference on Learning Representations*.

Ashwin Paranjape and Christopher Manning. 2021. Human-like informative conversations: Better acknowledgements using conditional mutual information. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 768–781, Online. Association for Computational Linguistics.

Ashwin Paranjape, Abigail See, Kathleen Kenealy, Haojun Li, Amelia Hardy, Peng Qi, Kaushik Ram Sadagopan, Nguyet Minh Phu, Dilara Soylu, and Christopher D Manning. 2020. Neural Generation Meets Real People: Towards Emotionally Engaging Mixed-Initiative Conversations. *Alexa Prize Proceedings 2020*.

Prasanna Parthasarathi and Joelle Pineau. 2018. Extending Neural Generative Conversational Model using External Knowledge Sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 690–695, Brussels, Belgium. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jairo Perez-Osorio and Agnieszka Wykowska. 2020. Adopting the intentional stance toward natural and artificial agents. *Philosophical Psychology*, 33(3):369–395.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a Benchmark for Knowledge Intensive Language Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Jan Pichi, Petr Marek, Jakub Konrád, Martin Matulık, and Jan Šedivy. 2018. Alquist 2.0: Alexa Prize socialbot based on sub-dialogue models. *Proc. Alexa Prize*.

Martin J. Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27(2):169–190.

Martin J. Pickering and Simon Garrod. 2006. Alignment as the Basis for Successful Communication. *Research on Language and Computation*, 4:203–228.

Anita Pomerantz. 1988. Offering a candidate answer: An information seeking strategy. *Communications Monographs*, 55(4):360–373.

Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, Bill Dolan, Yejin Choi, and Jianfeng Gao. 2019. Conversing by Reading: Contentful Neural Conversation with On-demand Machine Reading. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5427–5436, Florence, Italy. Association for Computational Linguistics.

Libo Qin, Tianbao Xie, Wanxiang Che, and Ting Liu. 2021. A Survey on Spoken Language Understanding: Recent Advances and New Frontiers. In *IJCAI*, pages 4577–4584.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Filip Radlinski and Nick Craswell. 2017. A Theoretical Framework for Conversational Search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, page 117–126, New York, NY, USA. Association for Computing Machinery.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21(1).

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anushree Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. 2017. Conversational AI: The science behind the Alexa Prize. In *Alexa Prize SocialBot Grand Challenge 1 Proceedings*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Rachel Reichman. 1985. *Getting Computers to Talk like You and Me*. MIT Press, Cambridge, MA, USA.

Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020. Thinking Globally, Acting Locally: Distantly Supervised Global-to-Local Knowledge Selection for Background Based Conversation. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, volume abs/1908.09528.

Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021. End-to-End Training of Neural Retrievers for Open-Domain Question Answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6648–6662, Online. Association for Computational Linguistics.

Harvey Sacks and Gail Jefferson. 1995. *Lectures on Conversation*. John Wiley & Sons, Ltd.

Chinnadhurai Sankar, Sandeep Subramanian, Christopher Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. *arXiv preprint arXiv:1906.01603*.

Skipper Seabold and Josef Perktold. 2010. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Abigail See and Christopher Manning. 2021. Understanding and predicting user dissatisfaction in a neural generative chatbot. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–12, Singapore and Online. Association for Computational Linguistics.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? How controllable attributes affect human judgments. In *Proceedings of NAACL-HLT*, pages 1702–1723.

Pararth Shah, Dilek Hakkani-Tür, Bing Liu, and Gokhan Tür. 2018. Bootstrapping a Neural Conversational Agent with Dialogue Self-Play, Crowdsourcing and On-Line Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51, New Orleans - Louisiana. Association for Computational Linguistics.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1577–1586, Beijing, China. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. *arXiv preprint arXiv:2104.07567*.

Jack Sidnell. 2016. Conversation Analysis. In *Oxford Research Encyclopedia of Linguistics*. Oxford University Press.

Vicki L Smith and Herbert H Clark. 1993. On the course of answering questions. *Journal of memory and language*, 32(1):25–38.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In *Proceedings of the 2015 Conference of the North American Chapter of* *the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado. Association for Computational Linguistics.

Robert Stalnaker. 2002. Common ground. *Linguistics and philosophy*, 25(5/6):701–721.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Kai Sun, Seungwhan Moon, Paul Crook, Stephen Roller, Becka Silvert, Bing Liu, Zhiguang Wang, Honglei Liu, Eunjoon Cho, and Claire Cardie. 2021. Adding Chit-Chat to Enhance Task-Oriented Dialogues. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1570–1583, Online. Association for Computational Linguistics.

Robert S. Taylor. 1968. Question-Negotiation and Information Seeking in Libraries. *College & Research Libraries*, 29(3):178–194.

Jaime Teevan, Kevyn Collins-Thompson, Ryen W. White, Susan Dumais, and Yubin Kim. 2013. Slow Search: Information Retrieval without Time Constraints. In *Proceedings of HCIR 2013*. ACM - Association for Computing Machinery.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Anu Venkatesh, Chandra Khatri, Ashwin Ram, Fenfei Guo, Raefer Gabriel, Ashish Nagar, Rohit Prasad, Ming Cheng, Behnam Hedayatnia, Angeliki Metallinou, et al. 2018. On evaluating and comparing open domain dialog systems. *arXiv preprint arXiv:1801.03625*.

Oriol Vinyals and Quoc V. Le. 2015. A Neural Conversational Model. *CoRR*, abs/1506.05869.

Ellen M. Voorhees. 2004. Overview of the TREC 2004 Robust Track. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004)*. NIST Special Publication SP 500-261.

Wayne Ward, Ronald Cole, Daniel Bolanos, Cindy Buchenroth-Martin, Edward Svirsky, Sarel Van Vuuren, Timothy Weston, Jing Zheng, and Lee Becker. 2011. My science tutor: A conversational multimedia virtual tutor for elementary school science. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(4):1–29.

David H.D. Warren and Fernando C.N. Pereira. 1982. An Efficient Easily Adaptable System for Interpreting Natural Language Queries. *American Journal of Computational Linguistics*, 8(3-4):110–122.

Joseph Weizenbaum. 1966. ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine. *Commun. ACM*, 9(1):36–45.

Ryen W. White. 2018. Skill Discovery in Virtual Assistants. *Commun. ACM*, 61(11):106–113.

Kathryn Whitenton and Raluca Budiu. 2018. The Paradox of Intelligent Assistants: Poor Usability, High Adoption. *Neilsen Norman Group Website*.

Jason Williams and Geoffrey Zweig. 2016. End-To-End LSTM-Based Dialog Control Optimized With Supervised And Reinforcement Learning. Technical Report MSR-TR-2016-72, Microsoft Research.

Terry Winograd. 1972. Understanding natural language. *Cognitive Psychology*, 3(1):1–191.

Jörg Wittwer, Matthias Nückles, and Alexander Renkl. 2008. Is underestimation less detrimental than overestimation? The impact of experts' beliefs about a layperson's knowledge on learning and question asking. *Instructional Science*, 36:27–52.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019a. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.03771.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019b. TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents. *CoRR*, abs/1901.08149.

Shali Wu and Boaz Keysar. 2007. The Effect of Information Overlap on Communication Effectiveness. *Cognitive science*, 31 1:169–81.

Jing Xu, Arthur Szlam, and Jason Weston. 2022. Beyond Goldfish Memory: Long-Term Open-Domain Conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5180–5197, Dublin, Ireland. Association for Computational Linguistics.

Michael Yeomans, Maurice E. Schweitzer, and Alison Wood Brooks. 2022. The Conversational Circumplex: Identifying, prioritizing, and pursuing informational and relational motives in conversation. *Current Opinion in Psychology*, 44:293–302.

Dian Yu, Michelle Cohn, Yi Mang Yang, Chun-Yen Chen, Weiming Wen, Jiaping Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhowmick, Shreenath Iyer, Giritheja Sreenivasulu, Sam Davidson, Ashwin Bhandare, and Zhou Yu. 2019. Gunrock: A Social Bot for Complex and Engaging Long Conversations. ArXiv preprint arXiv:1910.03042.

Dian Yu and Zhou Yu. 2019. MIDAS: A Dialog Act Annotation Scheme for Open Domain Human Machine Spoken Conversations. ArXiv preprint arXiv:1908.10023.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. KG-FiD: Infusing Knowledge Graph in Fusion-in-Decoder for Open-Domain Question Answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4961–4974, Dublin, Ireland. Association for Computational Linguistics.

Munazza Zaib, Wei Emma Zhang, Quan Z Sheng, Adnan Mahmood, and Yang Zhang. 2021. Conversational question answering: A survey. *arXiv preprint arXiv:2106.00874*.

Haolan Zhan, Lei Shen, Hongshen Chen, and Hainan Zhang. 2021. CoLV: A Collaborative Latent Variable Model for Knowledge-Grounded Dialogue Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2250–2261, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *ACL, system demonstration*.

Yizhe Zhang, Siqi Sun, Xiang Gao, Yuwei Fang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2022. RetGen: A Joint Framework for Retrieval and Grounded Text Generation Modeling. In *AAAI*.

Xinyan Zhao, Bin He, Yasheng Wang, Yitong Li, Fei Mi, Yajiao Liu, Xin Jiang, Qun Liu, and Huanhuan Chen. 2022. UniDS: A Unified Dialogue System for Chit-Chat and Task-oriented Dialogues. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 13–22, Dublin, Ireland. Association for Computational Linguistics.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-Grounded Dialogue Generation with Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390, Online. Association for Computational Linguistics.

Wen Zheng, Natasa Milic-Frayling, and Ke Zhou. 2020. Approximation of Response Knowledge Retrieval in Knowledge-grounded Dialogue Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3581–3591, Online. Association for Computational Linguistics.

Wen Zheng, Natasa Milic-Frayling, and Ke Zhou. 2021. Knowledge-Grounded Dialogue Generation with Term-level De-noising. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2972–2983, Online. Association for Computational Linguistics.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A Dataset for Document Grounded Conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713, Brussels, Belgium. Association for Computational Linguistics.

Pei Zhou, Karthik Gopalakrishnan, Behnam Hedayatnia, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. 2021. Commonsense-Focused Dialogues for Response Generation: An Empirical Study. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 121–132, Singapore and Online. Association for Computational Linguistics.

Chenguang Zhu, Yichong Xu, Xiang Ren, Bill Yuchen Lin, Meng Jiang, and Wenhao Yu. 2022. Knowledge-Augmented Methods for Natural Language Processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 12–20, Dublin, Ireland. Association for Computational Linguistics.