Neural Generation Meets Real People: Building a Social, Informative Open-Domain Dialogue Agent

Ethan A. Chi[*], **Ashwin Paranjape**[*], **Abigail See**[*], **Caleb Chiam**[*], **Trenton Chang**, **Kathleen Kenealy**, **Swee Kiat Lim**, **Amelia Hardy**, **Chetanya Rastogi**, **Haojun Li**, **Alexander Iyabor**, **Yutong He**, **Hari Sowrirajan**, **Peng Qi**, **Kaushik Ram Sadagopan**, **Nguyet Minh Phu**, **Dilara Soylu**, **Jillian Tang**, **Avanika Narayan**, **Giovanni Campagna**, and **Christopher D. Manning**

Stanford NLP
{ethanchi, ashwinp, abisee, calebc96, manning}@cs.stanford.edu

Abstract

We present Chirpy Cardinal, an open-domain social chatbot. Aiming to be both informative and conversational, our bot chats with users in an authentic, emotionally intelligent way. By integrating controlled neural generation with scaffolded, hand-written dialogue, we let both the user and bot take turns driving the conversation, producing an engaging and socially fluent experience. Deployed in the fourth iteration of the Alexa Prize Socialbot Grand Challenge, Chirpy Cardinal handled thousands of conversations per day, placing second out of nine bots with an average user rating of 3.58/5.

1 Introduction

Despite recent major advances (Adiwardana et al., 2020), open-domain *chit-chat*—friendly, social, casual conversation—remains a challenging task. In addition to difficulties with the sheer length and open-endedness required, social chatbots, or "socialbots," often struggle with *fluency*—whether due to the canned responses of manually constructed dialogue trees (Walker et al., 2001) or the anomalies of neural generators (Nie et al., 2021). But just being error-free isn't enough: to have a rewarding conversation, socialbots must be *personable*—displaying emotional intelligence, a rich personality, and an understanding of social dynamics. Although methods exist to address many of these issues individually, combining all of these features into a full-bodied conversation remains difficult.

In this paper, we describe Chirpy Cardinal, an open-domain conversational socialbot, which aims to bridge the gap between traditional dialogue tree-based approaches (Walker et al., 2001; Chen et al., 2018) and large pretrained neural dialogue agents (Adiwardana et al., 2020; Roller et al., 2020). Capable of discussing thousands of topics, Chirpy centers emotional and social intelligence with the goal of authentic, engaging interaction. Specifically, we make the following contributions:

- Conversations with open-domain socialbots often lack a stable structure. To ameliorate this, we present an **extensible design** for open-domain dialogue which prioritizes conversational stability and flexibility through mixed initiative (Horvitz, 1999).
- Although pretrained neural generators can be extremely fluent (Collins and Ghahramani, 2021), real-life deployment can suffer from a lack of both controllability and consistency (Nie et al., 2021). To address this, we describe several approaches to **integrate neural generation** into a symbolic setup, achieving local fluency without sacrificing global coherence.
- Towards the goal of a rewarding conversation, we suggest a set of approaches—ranging from small routines to complete submodules—which aim to make our socialbot a **good conversational partner**. We focus on being both *flexible*—handling a wide variety of topics in an interesting and informative way (Section 4)—and *personable*—empathizing with the other interlocutor even in difficult topics or situations (Section 5).

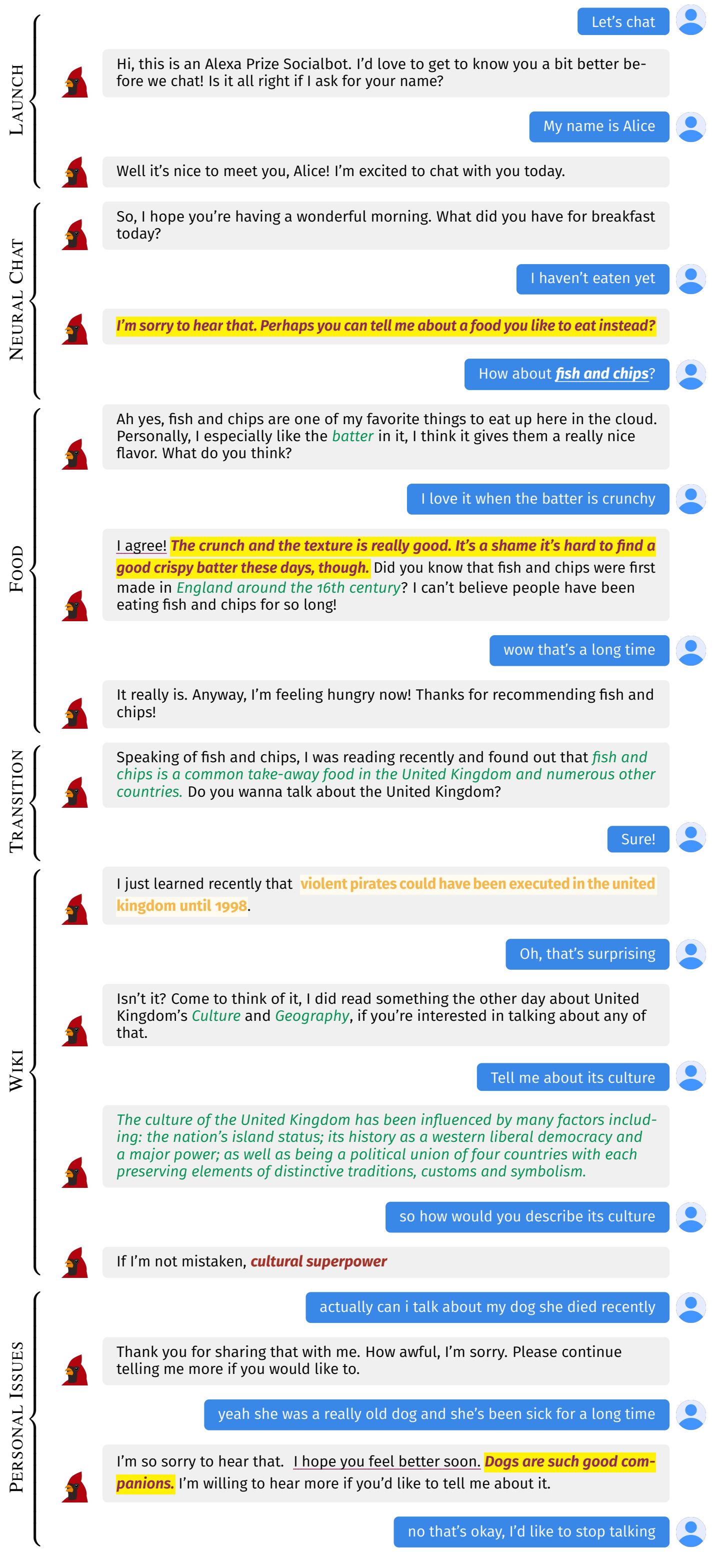
Deployed in the Alexa Prize Socialbot Grand Challenge 4, Chirpy Cardinal reached thousands of users per day; with conversations lasting up to 45 minutes at a time, it placed second out of nine agents in the finals. We open-source our system as an extensible framework for open-domain social dialogue,[1] providing an example of real-world deployment of conversational NLP systems.[2]

[*] Equal contribution.

[1] github.com/stanfordnlp/chirpycardinal

[2] stanfordnlp.github.io/chirpycardinal

Proceedings of the SIGdial 2022 Conference, pages 376–395
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022 ©2022 Association for Computational Linguistics



Let's chat

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 always begins the conversation. If we've talked with the user before, we confirm their identity (*"It's nice to chat with you again, Alice!"*).

My name is Alice

Well it's nice to meet you, Alice! I'm excited to chat with you today.

So, I hope you're having a wonderful morning. What did you have for breakfast today?

Neural Chat prompts first-time users about food, a subject of universal interest.

I haven't eaten yet

I'm sorry to hear that. Perhaps you can tell me about a food you like to eat instead?

Our neural generator responds directly.

How about ***fish and chips***?

Entity Linker (B) identifies ***fish and chips*** as the new entity.

Ah yes, fish and chips are one of my favorite things to eat up here in the cloud. Personally, I especially like the *batter* in it, I think it gives them a really nice flavor. What do you think?

With a food item mentioned, **Food** (E.5) starts. Phrases like *"in the cloud"* acknowledge our semi-embodied status.

I love it when the batter is crunchy

I agree! ***The crunch and the texture is really good. It's a shame it's hard to find a good crispy batter these days, though.*** Did you know that fish and chips were first made in *England around the 16th century*? I can't believe people have been eating fish and chips for so long!

Neural generation with the prefix "I agree!" produces a controllable opinion that doesn't contradict the user. We then add a fun fact scraped from Wikipedia.

wow that's a long time

It really is. Anyway, I'm feeling hungry now! Thanks for recommending fish and chips!

Food terminates; note that we respond *"It really is"* to any comments containing *"that's"*.

Speaking of fish and chips, I was reading recently and found out that *fish and chips is a common take-away food in the United Kingdom and numerous other countries.* Do you wanna talk about the United Kingdom?

We generate a **Transition** from *Fish and chips* to ***United Kingdom*** to continue the conversation smoothly.

Sure!

The user accepts the new entity.

I just learned recently that **violent pirates could have been executed in the united kingdom until 1998**.

Wiki (4.2) offers a **fun fact** from Reddit, rephrased for conversationality.

Oh, that's surprising

Isn't it? Come to think of it, I did read something the other day about United Kingdom's *Culture* and *Geography*, if you're interested in talking about any of that.

Wiki switches to a more informational mode, suggesting sections from the Wikipedia article to discuss.

Tell me about its culture

The culture of the United Kingdom has been influenced by many factors including: the nation's island status; its history as a western liberal democracy and a major power; as well as being a political union of four countries with each preserving elements of distinctive traditions, customs and symbolism.

Wiki gives a summary of the requested section.

so how would you describe its culture

Neural annotator detects question (D).

If I'm not mistaken, ***cultural superpower***

Wiki's ***Q&A module*** responds (C.4).

actually can i talk about my dog she died recently

User abruptly shares a personal issue.

Thank you for sharing that with me. How awful, I'm sorry. Please continue telling me more if you would like to.

Personal Issues provides emotional validation and encourages user sharing (5.1).

yeah she was a really old dog and she's been sick for a long time

User shares more details.

I'm so sorry to hear that. I hope you feel better soon. ***Dogs are such good companions.*** I'm willing to hear more if you'd like to tell me about it.

We acknowledge user sharing with a ***neural-generated*** response indicating willingness to listen further.

no that's okay, I'd like to stop talking

Figure 1: An example dialogue. To respect users' privacy, this is a conversation with an author, not a real user. We model dialogue as a series of subconversations (left side), whose ***topics*** are parsed and tracked by a neural entity linker. Each subconversation is managed by a RG, whose "scaffolded" architecture comprises hand-written treelets (plain text) incorporating numerous dynamic elements, ranging from ***neural generation*** to *retrieval from Wikipedia* to **neurally rephrased fun facts**. Prefix-based generation provides controllability, especially for sensitive topics like personal issues.

2 Design

2.1 System Design

We model a user dialogue as a series of subconversations (Figure 1), each handled by a *response generator* (RG). Varying greatly in scope and domain, each RG handles a specific topic (e.g. MOVIES, SPORTS) grounded in the outside world. RGs comprise dialog trees (Weizenbaum et al., 1966), whose tree nodes, which we term *treelets*, implement custom logic (e.g. intent classification or retrieval) to generate a response.

At the start of each turn, the user utterance is annotated for linguistic features (Appendix C), then processed in parallel by all RGs. By default, the previous turn's RG is selected; should the RG that last responded crash or a different RG request to take over, we seamlessly switch RGs and move to a new subconversation.

2.2 Navigation

To enable mixed initiative—shared user-bot responsibility in driving the conversation (Horvitz, 1999)—we provide a suggested navigational path, while letting users deviate drastically from it. Specifically, each RG continues through its dialogue tree until exhausting its subconversation; we then transition to another RG by bringing up a previously user-mentioned topic (*"You mentioned **cats** earlier; would you say you're a big fan?"*), mentioning a tangentially related topic that we can discuss well, or simply sampling a new RG and corresponding topic at random. Users may explicitly change the topic (*"can we talk about roblox"*); implicitly suggest a desire to redirect the conversation (*"yeah"* or *"uh-huh"*); or otherwise behave in ways that require the bot to act dynamically (*"i don't know, how about you?"*). We handle these deviations from the conversational flow through neural handlers that allow periods of flexibility before returning to the overall conversational structure (Appendix F).

2.3 Entity Handling

To allow users to discuss a vast array of interesting topics relevant to their lives, we support any Wikipedia entity as a topic of discussion.[3] To do so, we entity-link (Kolitsas et al., 2018) the user utterance to relevant entities using a fine-tuned BERT model (Broscheit, 2019; also B.3), mitigating ASR errors through a phonetic similarity search (B.2). Since incorporating Wikipedia article titles directly into bot utterances can be awkward (e.g. *"can we talk about **cat**"*), we refer to entities by more natural *talkable names* (e.g. *"**cats**"*), generated using GPT-3 (Brown et al., 2020).

RG	Prefix	Sample Completion
FOOD	A hoagie is a great choice! I especially love. . .	*"...mine with a little cheese and bacon!"*
PERSONAL ISSUES	That sounds frustrating. I hope that. . .	*"...she feels better soon."*

Table 1: Sample uses of conditional neural generation.

3 Neural Generation

Although neural generative models (Roller et al., 2021) have achieved success in open-domain dialogue, significant obstacles impede deployment in real-life situations: neural text degeneration (Holtzman et al., 2020; Welleck et al., 2019), hallucination (Dziri et al., 2021), and inconsistency (Zhang et al., 2018), In addition, large latency can make models challenging to deploy in practice (Worswick, 2020). In this section, we investigate ways to utilize the power of such models in the context of structured dialogue. We propose integrating neural generation in the context of hand-written scaffolding, aiming to benefit from its variety and fluency while maintaining coherency over time.

3.1 DistillBlender: A Fast, General-Purpose Neural Generator

For general use, we distill a single model from BlenderBot-3B (Roller et al., 2021) with 9 decoder layers,[4] reducing latency significantly over the original model. We use it as follows:

- The **NEURAL CHAT** RG, which directly exposes lightly edited neural model outputs as a subconversation. Due to BlenderBot's end-to-end training, this is initially a rich, fluent conversational experience, but due to rapid degradation we terminate after 5 turns.
- *Conditional prompting* (Keskar et al., 2019), which enables controllability in a structured context. We apply hand-written prefixes to guide the model towards fluent, contextually appropriate completions (Table 1).

[3] Specifically, those with sufficiently high cross-references and meeting certain criteria for definiteness (Appendix B.1).

[4] Reduced from an original 24.

Template	I love how **[actor]** acted in **[film]**, especially their **<mask>**.
Infilled	I love how **[Keanu Reeves]** acted in **[The Matrix]**, especially their **ability to freeze time**.

Table 2: An example of template-based infilling using Keanu Reeves as the knowledge source.

3.2 Template-Based Infilling

Towards the goal of rich, coherent conversation for a wide class of topics, we propose *template-based infilling*, a more flexible version of standard slot-filling methods (Haihong et al., 2019) that does not require a structured knowledge base. Using both freeform information and an end-user-defined template, we use a fine-tuned BART model (Lewis et al., 2020) to generate a grounded statement. Defining a diverse set of templates for each entity category allows us to provide expressive yet controllable conversation on many different types of entities (Table 2).

4 Response Generators

4.1 News

The **News** RG aims to discuss current events, which often feature heavily in typical human-to-human chit-chat (De Boer and Velthuijsen, 2001). When an entity or topic mentioned in *The Washington Post* or *The Guardian* appears in conversation, we offer a headline, conversationally paraphrased using GPT-3 (Brown et al., 2020), as a subject of conversation.[5] If the user is interested, we provide a summarized (Zhang et al., 2020a) snippet of the storyand allow the user to ask follow-up questions answered via neural QA (Clark et al., 2020; Rajpurkar et al., 2018; also B.3). Answers are then rephrased (Paranjape et al., 2020) and reranked using PCMI (Paranjape and Manning, 2021), allowing our socialbot to dynamically integrate current events into conversations when relevant.

4.2 Wiki

In contrast to humans, open-domain chatbots are commonly expected to be able to "engage in conversation on any topic" (Adiwardana et al., 2020). Towards this end, the **Wiki** RG discusses any entity. We aim to be informative, not overwhelming; in addition to encouraging users to share their own knowledge and experience about the entity, we bring up interesting factoids from /r/todayilearned (conversationally rephrased; E.3.4), as well as infilled remarks. We then discuss the entity in more depth based on its article, flexibly acknowledging user questions and comments with the Q&A handler (C.4) or neural generation.

4.3 Opinion

A core part of social chit-chat (Walker, 2009), exchanging and commenting on opinions allows a socialbot to project a stronger sense of personality. The **Opinion** RG solicits users' opinions on topics and reciprocates with its 'own' opinions (sourced from Twitter), including occasional *disagreement* to help engage user interest (E.4).

4.4 Rules-based RGs

In order to broaden the scope of our bot, we manually build several domain-specific response generators. **Food**, which always opens the conversation, discusses common foods scraped from Wikipedia.[6] **Movies** uses the Alexa Linked Data API to discuss movies and actors. **Music** uses the MusicBrainz[7] database to discuss songs, artists, and music genres. **Sports** uses the ESPN API to discuss NFL football and NBA basketball. We describe these RGs in more detail in Appendix E.

5 Being Personable

To achieve truly social conversation, a socialbot must be a *good conversational partner*: empathetic, supportive, and interested in what its human interlocutor has to say (Salovey and Mayer, 1990; Li et al., 2017). In this section, we describe several approaches that aim to achieve this, ranging from full RGs to smaller subroutines.

5.1 Handling Personal Issues

Many users—especially those who chat with our socialbot looking for companionship—share personal struggles with our bot, requiring emotional sensitivity and tact. Handling such conversations purely neurally would result in rapid degeneration due to neural toxicity (Dinan et al., 2021). To address this, the Personal Issues RG responds to personal disclosures using active listening techniques (Bodie et al., 2015), asking exploratory questions about

[5] We use `davinci` with the following prompt: "Paraphrase news headlines into a complete, grammatical sentence in plain English. The sentence should be in the past tense."

[6] In practice, we found that always starting with Food proved to be most successful for ratings (E.1), perhaps since food is such a universal human need and discussion point.

[7] `https://musicbrainz.org/`

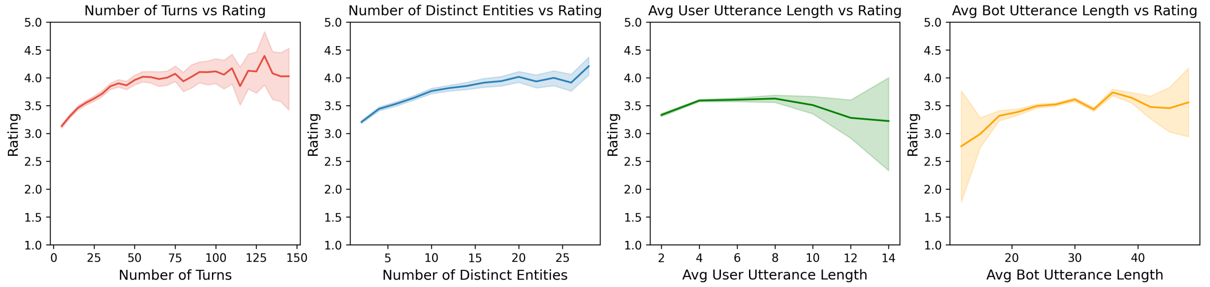


Figure 2: Engagement metrics vs. rating. We bucket (with size 5, 2, 2, 3 respectively) conversations based on four engagement metrics—number of turns, number of distinct entities, average user utterance length, and average both utterance length—and plot each bucket against user rating (Likert 1-5 scale, measured per-conversation). 95% confidence intervals computed via bootstrapping ($n = 1000$).

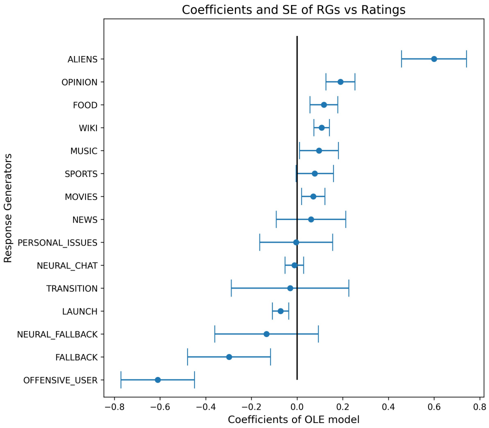


Figure 3: Linear regression coefficients for response generator vs. rating; each RG is weighted by the number of turns it contributes. 95% confidence intervals determined via bootstrapping with $n = 1000$.

the nature of the user's issue (*"When did you start feeling this way?"*), and validating their concerns (*"I see, that sounds difficult."*)

On the other hand, a significant subset of users become verbally abusive during the conversation (Curry and Rieser, 2018, 2019). We follow the strategy of Li et al. (2021): a de-escalating statement to avoid confrontation, addressing the user by name (*"John"*); then changing the topic.

5.2 Self-disclosure

The ALIENS RG allows the socialbot to muse about its pet topic—the possible existence of extraterrestrial life—as well as its own identity and sense of purpose. Contrasting with purely informational modes, this RG fleshes out a personality for our agent and enables *self-disclosure*—disclosing goals, attitudes, and personal interests to support interpersonal intimacy (Altman and Taylor, 1973; Ignatius and Kokkonen, 2007).[8]

5.3 Personalization

Users often expect chatbots to remember personal preferences and user details (Chaves and Gerosa, 2021; Svikhnushina et al., 2021) and to tailor their responses accordingly (Neururer et al., 2018; Shum et al., 2018). We personalize bot responses with the user's preferences: for example, in regards to the Olympics, *"Ah, that makes sense since you did say it's your favorite sport!"*. Referencing this user state across conversations makes repeated conversations with Chirpy feel fresh and dynamic, rather than retreading past questions and topics.

6 Results

In this work, we have outlined a set of design priorities and corresponding approaches to design a fluent, flexible, and sociable chatbot. We validate these through the Alexa Socialbot Grand Challenge 4: engaging in approximately 1,000 conversations per day, our socialbot achieved an average user rating of **3.55**, ending the development period tied for first place in rating.[9] Validating our design goals, we observe high ratings for a hybrid neural-scaffolded approach (FOOD, etc.), personable RGs (ALIENS), and open-domain techniques (WIKI) (Figure 3). Our socialbot engages in long, varied conversations without repeating itself (Figure 2).

That said, both overall rating and sample conversations testify that Chirpy remains far from the goal of truly compelling and enjoyable human-bot interaction. We do not argue that our approaches are sufficient—or even necessary—to create such an ideal system; rather, we hope that the *priorities* outlined here can serve as a starting point to help inform further socialbot development, whether purely neural or hybrid in nature.

[8] This RG comes up only after sufficient rapport has been built—i.e. after 30 turns in the conversation.

[9] Likert scale between 1 and 5; overall average across teams was **3.47**. For more information, please consult the proceedings of the Alexa Prize Socialbot Grand Challenge 4.

Ethics Statement

In this work, we have presented a conversational agent that conducts an open-domain dialogue. We believe that many people would enjoy having a chat partner who is empathetic and knowledgeable, and our ratings seem to suggest that a reasonable number of people appreciate their conversations enough to want to talk to the bot again. Prior to engaging with the chatbot, all user participants are required to consent to their conversations, feedback, and ratings being recorded, as per the Alexa Terms of Use. Additionally, the chatbot clearly identifies itself as a bot at the start of each conversation. No actual user conversations or identifying information are used in this paper.

However, as our system incorporates computational methods for generating conversational utterances automatically, there exists a risk that users may be exposed to unsafe utterances or discussion topics. Conversational models of all kinds can produce sexist, racist, or otherwise unsafe statements; neural conversational agents can be particularly vulnerable due to pre-training on Internet chat forums, which can be particularly toxic (Xu et al., 2020). Towards this end, our system incorporates a safety module that prevents our model from producing utterances with certain hard-coded words or categories. Yet the use of a blacklist in itself raises additional ethical issues, as poorly designed blacklists can marginalize communities by blocking topics that ideally, one should be able to discuss equitably.

Finally, the human-like nature of open-domain dialogue systems can be particularly damaging when used in an adversarial context, e.g. by state actors (Boshmaf et al., 2012). Ultimately, like all text generation methods, the benefit of releasing an open-domain dialogue model must be weighed against its possible downsides.

Acknowledgements

We thank Amazon.com, Inc. for a grant partially supporting the work of the team and *The Guardian* for allowing us to use their news API for our system. Additionally, we thank Anna Goldie and Monica Lam for helpful discussions.

The user icon in Figure 1 is from *kdg design* and used under a free license.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Irwin Altman and Dalmas A Taylor. 1973. *Social penetration: The development of interpersonal relationships.* Holt, Rinehart & Winston.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of the 5th International Conference on Learning Representations*.

Graham D Bodie, Andrea J Vickery, Kaitlin Cannava, and Susanne M Jones. 2015. The role of "active listening" in informal helping conversations: Impact on perceptions of listener helpfulness, sensitivity, and supportiveness and discloser emotional improvement. *Western Journal of Communication*, 79(2):151–173.

Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. 2012. Key challenges in defending against malicious socialbots. In *5th {USENIX} Workshop on Large-Scale Exploits and Emergent Threats ({LEET} 12)*.

Samuel Broscheit. 2019. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685, Hong Kong, China. Association for Computational Linguistics.

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.

Ana Paula Chaves and Marco Aurelio Gerosa. 2021. How should my chatbot interact? a survey on social characteristics in human–chatbot interaction design. *International Journal of Human–Computer Interaction*, 37(8):729–758.

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*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Eli Collins and Zoubin Ghahramani. 2021. LaMDA: our breakthrough conversation technology. *Google AI Blog*.

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.

Connie De Boer and Aart S Velthuijsen. 2001. Participation in conversations about the news. *International Journal of Public Opinion Research*, 13(2):140–158.

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, Gavin Abercrombie, A Stevie Bergman, Shannon Spruit, Dirk Hovy, Y-Lan Boureau, and Verena Rieser. 2021. Anticipating safety issues in e2e conversational ai: Framework and tooling. *arXiv preprint arXiv:2107.03451*.

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 *Proceedings of the International Conference on Learning Representations (ICLR)*.

Nouha Dziri, Andrea Madotto, Osmar Zaiane, and Avishek Joey Bose. 2021. Neural path hunter: Reducing hallucination in dialogue systems via path grounding. *arXiv preprint arXiv:2104.08455*.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking training with large-scale human feedback data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 386–395, Online. Association for Computational Linguistics.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinglang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, Dilek Hakkani-Tür, and Amazon Alexa AI. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. In *INTERSPEECH*, pages 1891–1895.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. 2015. Deep learning with limited numerical precision. In *International conference on machine learning*, pages 1737–1746. PMLR.

E Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Emmi Ignatius and Marja Kokkonen. 2007. Factors contributing to verbal self-disclosure. *Nordic Psychology*, 59(4):362–391.

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*.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2020. Deep encoder, shallow decoder: Reevaluating the speed-quality tradeoff in machine translation. *CoRR*, abs/2006.10369.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

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*.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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.

Haojun Li, Dilara Soylu, and Christopher D. 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*. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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.

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.

Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. 2020. How decoding strategies affect the verifiability of generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 223–235, Online. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 152–159.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

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

Mario Neururer, Stephan Schlögl, Luisa Brinkschulte, and Aleksander Groth. 2018. Perceptions on authenticity in chat bots. *Multimodal Technologies and Interaction*, 2(3):60.

Yixin Nie, Mary Williamson, Mohit Bansal, Douwe Kiela, and Jason Weston. 2021. I like fish, especially dolphins: Addressing contradictions in dialogue modeling. 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 1699–1713.

Ashwin Paranjape and Christopher Manning. 2021. Human-like informative conversations: Better acknowledgements using conditional mutual information. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 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. *arXiv preprint arXiv:2008.12348*.

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

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.

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.

Stephen Roller, Y-Lan Boureau, Jason Weston, Antoine Bordes, Emily Dinan, Angela Fan, David Gunning, Da Ju, Margaret Li, Spencer Poff, et al. 2020. Open-domain conversational agents: Current progress, open problems, and future directions. *arXiv preprint arXiv:2006.12442*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Peter Salovey and John D Mayer. 1990. Emotional intelligence. *Imagination, cognition and personality*, 9(3):185–211.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *CoRR*, abs/1804.04235.

Sam Shleifer and Alexander M. Rush. 2020. Pre-trained summarization distillation. ArXiv preprint arXiv:2010.13002.

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

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030, Online. Association for Computational Linguistics.

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):339–373.

Ekaterina Svikhnushina, Alexandru Placinta, and Pearl Pu. 2021. User expectations of conversational chatbots based on online reviews. In *Designing Interactive Systems Conference 2021*, pages 1481–1491.

Marilyn Walker, Rebecca J Passonneau, and Julie E Boland. 2001. Quantitative and qualitative evaluation of DARPA communicator spoken dialogue systems. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 515–522.

Marilyn A Walker. 2009. Endowing virtual characters with expressive conversational skills. In *International workshop on intelligent virtual agents*, pages 1–2. Springer.

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

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Steve Worswick. 2020. Bot battle update — we won?

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots. *arXiv preprint arXiv:2010.07079*.

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. 2021. MIDAS: A dialog act annotation scheme for open domain HumanMachine spoken conversations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1103–1120, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

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? *arXiv preprint arXiv:1801.07243*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020b. DialoGPT: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.

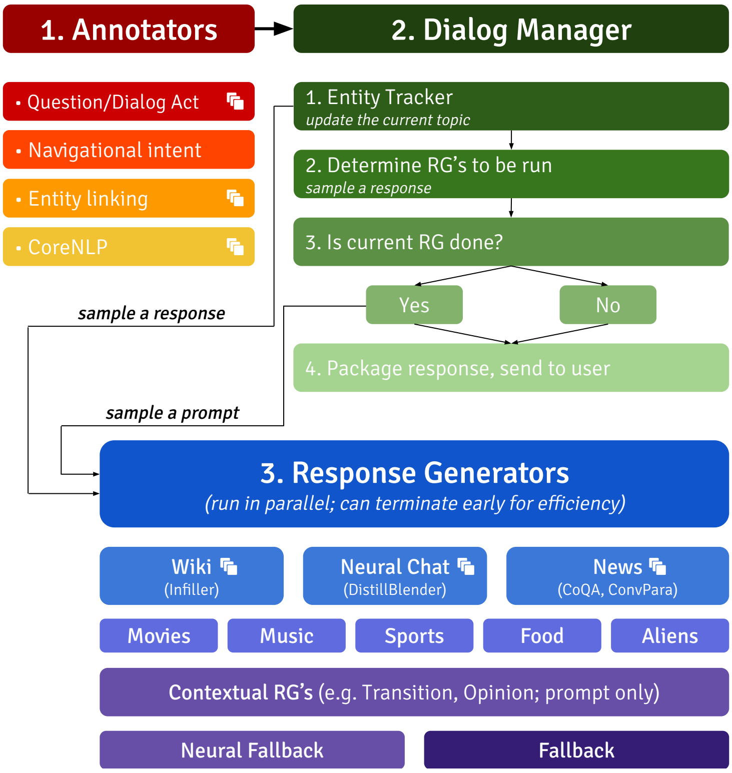


Figure 4: Overall system design.

A Additional Architectural Details

A.1 Overall Architecture

Our system (Figure 4) is based on CoBot (Khatri et al., 2018). During the Alexa Prize, Chirpy Cardinal ran on AWS Lambda, a serverless computing platform; our open-source demo runs on Kubernetes. For reliability, our function is stateless; therefore, to preserve information between turns, we store our bot's overall state in an external PostgreSQL state table (see Figure 4). We execute the following steps on each turn:

1. Fetch the previous turn's state from the state table.
2. Generate a response from our neural generator (for latency reasons; D.1).
3. Execute all annotators (C), which run on remote CPU-only instances.
4. Analyze the user utterance for **navigational intent** (A.3) to determine whether we should change topic.
5. Analyze the user utterance for entities (B.4). If warranted by the user's navigational intent or the last bot response, the **current entity** (B.4) is updated.
6. Run all RG's (Section 4) in parallel; RG's that require a neural response await the neural generator. Out of all received responses, select a response (A.2), and update the current entity if necessary.
7. If the chosen response generator has finished its conversation, we run our collection of RG's a second time to produce prompts (A.2) Select a prompt, update the current entity again if needed, and form the bot's utterance 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 RG. Each individual RG state contains information required by that RG – for example, it might contain the current treelet in the RG's dialogue graph, or a list of the utterances and/or entities that have been discussed, to avoid repetition.

A.2 Response Design

Responses and prompts both carry a *priority*, with the highest-priority response/prompt chosen at the corresponding stage. In general, the RG which responded last has the highest priority; however, RG's can optionally specify a lower priority so that other RG's take over, or a higher priority to take over from another RG. In practice, these priority levels are rarely used due to their tendency to produce a choppy conversation.

A.3 Navigational Intent Classifier

A user has *positive* navigational intent if they want to discuss a topic; conversely, *negative* navigational intent means that the user would like to avoid discussing a topic. Users may express navigational intent while specifying a topic (*"can we talk about minecraft"*, referring to the current topic (*"let's discuss this more"*), or referring to no topic (*"I don't want to chat anymore"*). Positive and negative navigational intents can even be combined (*"I don't want to talk about movies any more, let's chat about you"*). We classify use manually-constructed regexes, which achieve extremely high precision.

B Entity-Linking Details

Detecting and understanding references to real-world entities is essential to any open-domain conversational system; we find that users appreciate being able to discuss a wide variety of topics that interest them or are relevant to their lives. For our socialbot, we train and deploy a neural entity linker that links spans to Wikipedia entities.

B.1 Entity Pool

To obtain our pool of potential entities, we process the May 20th, 2020 dump of English-language Wikipedia[10] using MWParserFromHell[11] and Spark[12]. We store our data in a large ElasticSearch index, keeping only entities with at least 200 cross-references in Wikipedia. In total, we have $171,961$ entities.

Notably, certain entities are inappropriate to discuss even if correctly entity-linked by our model; for example, our system is unable to handle abstract nouns well (e.g., *philosophy*, *film*). To ameliorate this, we manually created a set of *low-precision* entities composed of both WikiData categories (e.g., *conspiracy theory*, *financial risk*, *research method*) and specific common entity names (e.g., *bank*, *catalog*, *coast*). The bot will not start a conversation itself about such entities; however, it is able to handle explicit user navigational requests (e.g., *can we talk about the bank*). Separately, we also ban certain racial, religious, and other identity-based terms that are unlikely to result in a good conversation on either the bot's or user's part, as well as certain short acronyms (e.g. *cet*, *ep*, *fm*) that are almost always triggered by ASR errors.

B.2 Candidate generation

For a given user utterance, we want to compute the set of entities that the user could possibly be referring to; for example, if the user mentions *"swift"*, this could refer to the bird, musical artist, or programming language. To do so, for each possible span, we pre-compute the set of entities for which the span serves as a Wikipedia anchortext, creating a mapping from spans to sets of candidate entities. At execution time, for all n-grams in the user utterance with 5 or fewer tokens[13], we retrieve the set of candidate entities from our database.

Since we do not have access to original user audio, ASR errors can impede candidate generation (Chen et al., 2018). For example, if an user's reference to the film *Ford v Ferrari* is erroneously transcribed as *"four v ferrari"*, a naïve entity linker will fail to identify the correct entity. To address this, we pre-compute phoneme and metaphone representations for all of our entities (e.g. converting *Harry Potter* to `'HH EH R IY P AA T ER'`[14] and `'HRPTR'`[15]). At execution time, each n-gram's candidate set is augmented with the sets for spans with similar phoneme/metaphone representations.

B.3 Entity disambiguation

Given a set of candidate entities, we want to select those candidates that the user is interested in. Towards this end, we fine-tune a BERT-medium (Devlin et al., 2019) to disambiguate entities, following Broscheit (2019) with minor modifications. Specifically, we learn an embedding for each entity in our dataset. Then given a span within an user utterance, we model the probability that the span refers to a given candidate entity as the dot product between the contextual span representation and the entity's embedding. At deployment, we only take entities with a predicted likelihood of at least 0.5; additionally, we use only the highest-likelihood entity for each span.

We depart from Broscheit by mean-pooling over the contextualized span representation, rather than doing per-token entity-level disambiguation. Fine-tuning takes about 20 days using 4 Titan X GPUs; during deployment, we execute using CPU only.

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

[11] https://mwparserfromhell.readthedocs.io/en/latest

[12] https://spark.apache.org

[13] specifically, those not solely composed of stopwords

[14] https://pypi.org/project/g2p-en/

[15] https://pypi.org/project/metaphone/

B.4 Entity Tracking

At any given point, we track the *current entity* (the current subject of conversation), a set of *untalked* entities (entities which the user has mentioned but we have not yet addressed), and a set of *rejected* entities (which the user does not want to discuss; these are no longer brought up by our bot.). These are updated every turn as follows:

- Entities receiving negative navigational intent (*"can we not talk about paraguay"*) are **rejected**. Non-specific negative navigational intent (*"let's not discuss this"*) causes the current entity to be rejected instead.
- Entities receiving positive navigational intent (*"can we talk about mexico"*) are **set as the current entity**. The previous conversation ends, with all RGs are prompted to handle this new current entity instead.
- If the currently active RG asked a question on the last turn, the current highest-priority entity is identified as the presumable user answer and **set as the current entity**. Additionally, if the previous question expects a particular category of entities (e.g. *"What's your favorite movie?"*), we pick the highest-priority entity matching the expected category (e.g., `film`).
- All remaining entities are marked as *untalked* (to be possibly discussed later).

C Annotators

All annotators—modules which provide linguistic annotations for the user utterance—are executed in parallel at the beginning of each turn.

C.1 CoreNLP

We use the following annotators from Stanford CoreNLP (Manning et al., 2014): tokenization, sentence splitting, part-of-speech tagging, lemmatization, named entity recognition, constituency parsing, dependency parsing, coreference resolution, and sentiment analysis. Due to the format of the user utterances (lowercase with no punctuation), we use caseless models[16] for part-of-speech tagging, constituency parsing and named entity recognition. We use these annotations for certain handwritten NLU operations.

Training Regime	Silver	Gold	Test F1
Baseline	0	0	0.53
Self-training ($\tau = .95$)	41,152	0	0.54
Self-training ($\tau = .75$)	62,150	0	0.54
Hand-labeled	0	2,407	**0.81**

Table 3: Performance of our Dialogue Act model under different training regimes. All models have access to $10,090$ examples in the MIDAS training set, but training a baseline model solely on these examples suffers from domain shift. *Self-training*, which first uses this baseline model to silver-label a large number of unlabeled Chirpy Cardinal examples with confidence above some cutoff τ, then retrains on the union of the two, does not improve performance. *Hand-labelling* a small amount of additional data significantly improves performance.

C.2 Dialogue Act Classifier

Dialogue acts, an ontology over user intents (Stolcke et al., 2000; Jurafsky et al., 1997), have been successfully employed in open-domain dialogue agents (Yu et al., 2019). We modify MIDAS (Yu and Yu, 2021)—an annotation schema designed specifically for human-chatbot dialogue— to better fit the needs of our bot, removing 4 labels[17] due to low frequency in our conversations and creating 5 new labels: *correction*, *clarification*, *uncertain*, *non-compliant*, and *personal question*. In total, our modified schema has 24 labels.

Evaluated on the MIDAS test set, a fine-tuned BERT baseline achieves .78 micro-F1; however, evaluated on an OOD test set composed of our own conversations, it achieves only .53 (Table 3). Although self-training (McClosky et al., 2006) proved ineffective, hand-labeling additional OOD conversations achieved a micro-F1 of 0.81. The predictions of this final model inform navigation, as well as RG-specific NLU.

C.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 ASR-transcribed user utterances do not contain punctuation. To recognize questions, we fine-tuned a RoBERTa model (Liu et al., 2019; Wolf et al., 2019) on an simplified version of the Dialogue Act training data, framing the task as binary classifica-

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

[17] *apology*, *apology-response*, *other*, and *thanks*

tion, conditioned only on the user utterance. This model achieved an F1-score of 0.92 and improved the reliability of question detection.

C.4 QA Annotator

The **QA annotator**, an ELECTRA-Large model (Clark et al., 2020) pretrained on SQuAD2.0 (Rajpurkar et al., 2018), performs question answering for the NEWS (Section 4.1) and WIKI (Section 4.2) RGs. Unlike other annotators, this annotator does not run unless called by these RGs.

D Neural Generation

Our neural agent is a distilled (Hinton et al., 2015) version of BlenderBot-3B (Roller et al., 2021), an autoregressive Seq2Seq model trained on Blended Skill Talk (Smith et al., 2020), Wizard of Wikipedia (Dinan et al., 2019b), ConvAI2 (Dinan et al., 2019a), and Empathetic Dialogues (Rashkin et al., 2019). We distill using Sanh et al. (2019)'s method (as implemented in ParlAI; Miller et al., 2017), using Adafactor (Shazeer and Stern, 2018) with learning rate 6.25×10^{-5}, validation loss-based LR reduction, warmup, and FP16 (Gupta et al., 2015). We used a batch size of 1 for training on a single V100 GPU.

For decoding, we use top-k sampling ($k = 5$) with temperature $T = 0.7$. To encourage response diversity across the conversation, we sample sequences of minimum length randomly chosen from 5, 10, 15, 20, 25; in practice, the length of the generations is 0-2 tokens above the minimum selected length. Additionally, we use delayed beam search (Massarelli et al., 2020), with the conversational history up to 128 tokens in the past serving as context. After decoding, we first filter out offensive, null, and repetitive responses, as well as questions after the first turn. We then select a final response based on the posterior likelihood, among other metrics.

D.1 Analysis

We find that our model qualitatively outperforms a GPT-2 (Radford et al., 2019) baseline fine-tuned on Empathetic Dialogues (Table 4), with similar latency. That said, our model still suffers certain limitations out-of-the-box; we discuss strategies for mitigating these issues.

Diversity-coherence tradeoff For our model, beam search decoding yields coherent but non-diverse responses, while stochastic decoding results in nonsensical generations even under top-*p* (Holtzman et al., 2020) or top-*k* (Fan et al., 2018) sampling. Delayed beam search, which samples the first few tokens before defaulting to beam search, yielded more stable behavior than stochastic decoding, and better diversity than beam search.

Degeneration The model outputs conversation-ending phrases (e.g., "I have to go", "It was nice talking") after 7 turns, hurting user experience. Manual examination of the training data revealed this is due to the short lengths of conversations in the training data collected via crowdworkers. We alleviated this with a manual blacklist of conversation-ending phrases and forcing NEURAL CHAT (Section 3.1) to hand off to another RG prior to the 7-turn limit.

Changing the topic Our model tends to generate one or more follow-up questions in every utterance, which are often unrelated to the current topic, creating a disorienting topic change (e.g. *"What's your favorite color?"*). To address this, NEURAL CHAT limits the number of questions in each neural chat *conversation* to one, with subsequent questions truncated. During conditional generation, questions are deleted altogether.

Latency Generating responses can take up to 2s in the worst-case, averaging just under 1s. To mitigate this, we **pre-fetch** responses from DistillBlender at the start of every turn, instead of waiting for the Neural Chat RG to request a neural response. If the responding RG returns a response without utilizing the DistillBlender response during the turn, we ignore its output and return early. Although this results in unused pings to the neural model, we find that the trade-off is worth it.

E Additional RG Details

E.1 Launch

At the beginning of the conversation, the bot initially selected icebreakers at random. However, we eventually found that certain icebreakers tended to fare better than others. Specifically, conversations starting with food-related icebreakers (e.g. "Do you have any recommendations for what I should cook at home?") had an average rating was 3.49 over a sample of 1405 conversations, compared to an average rating of 3.43 for non-food-related icebreakers (e.g. "What did you do over the weekend?") over a sample of 1418 conversations. Digging deeper,

Prev. bot utterance	User utterance	Blender (distilled)	GPT-2 baseline
I love pizza, what's your favorite toppings?	Pepperoni.	That's a good choice. Do you have a favorite kind of cheese? Mine is mozzarella.	Oh yea I love that pizza too. I don't like too much other toppings.
What did you do today?	Alexa, what's up?	Nothing much. I just got back from a run. I'm exhausted.	I'm sorry to hear that. What happened?
Have you been doing anything productive?	Well, I've been trying to be productive but I keep on breaking things.	Breaking things can be very frustrating. I hope you are able to get it all sorted out.	That's no good, it's not productive to be stuck on something and not get to something else.

Table 4: Responses from our neural generator, distilled from BlenderBot-3B (punctuation and capitalization added for clarity), compared to a GPT-2 baseline fine-tuned on Empathetic Dialogues. We find that our neural generator provides stronger performance with similar latency.

we found that if the second turn is handled by the Food RG, we achieved an average rating of 3.64 over 606 conversations, compared to an average rating of 3.49 if the second turn is handled by the Neural Chat RG, over 1684 conversations (second turns are mainly handled by Food and Neural Chat RG's, but sometimes by others).

This prompted us to update our Launch RG so that we open with a food-related question for all conversations, hence increasing the frequency of handing over to the Food RG.

E.2 News

The NEWS RG (Section 4.1) curates global news from The Washington Post[18] and The Guardian[19]. Article titles, topic categories, body texts, dates, and content URLs are stored in a constantly updating ElasticSearch index. When a topic or entity available in our index appears in conversation, the News RG brings up related stories from our database. In addition, NEWS also initiaties conversations about currently trending news topics by scraping trending news from Google Trends[20].

Behavior To produce a prompt usable in conversation, we rephrase the headline to conversational form using GPT-3 davinci-instruct-beta.[21] If the user expresses interest in continuing the conversation, the we provides a conversational summary generated by Pegasus-Multinews (Zhang et al., 2020a; Fabbri et al., 2019). Summaries are decoded using 8 beams and a maximum of 50 tokens for conversationality, and are pre-generated for efficiency; if the neural module fails, we instead use an extractive summary (Mihalcea and Tarau, 2004).

Follow-up If the user continues to be engaged, we prompt for questions or comments. If a comment is detected, a neural response is generated using a set of hand-written prefixes; If a question is detected (C.3), they are answered via the QA annotator (C.4). We then conversationally paraphrase the answer using a GPT-2-medium model (Radford et al., 2019) fine-tuned on Topical Chat (Gopalakrishnan et al., 2019) to produce a more human-like response. We use the truncated conversational history as the input history and a merged representation of the answer and the span as the the factual content. It outputs a conversational-sounding paraphrase of the answer. Finally, we rank the generated paraphrases using Fused-PCMI (Paranjape and Manning, 2021).

E.3 Wiki

To support our goal of high-coverage world knowledge (Section 1), the Wiki RG uses Wikipedia articles as grounding to discuss any entity that interests the user and that is not handled by any other RG. Our goal is to allow the user to conversationally discover interesting information about the entity.

E.3.1 Data

We use the Wikipedia dump from May 20th, 2020[22], processed using MWParserFromHell[23] and Spark.[24] We store our data in a large ElasticSearch

[18] https://washingtonpost.com

[19] https://theguardian.com

[20] https://trends.google.com

[21] We use the following prompt: "Paraphrase news headlines into a complete, grammatical sentence in plain English. The sentence should be in the past tense."

[22] https://dumps.wikimedia.org/backup-index.html

[23] https://mwparserfromhell.readthedocs.io/en/latest

[24] https://spark.apache.org

index.

E.3.2 Behavior

Wiki RG facilitates a discussion about an entity based on how it came up in conversation (see Fig. 5). If the user initiates an discussion about an entity, the RG encourages the user to share their own knowledge and experience about the entity. Otherwise, if the entity came up only in passing or as a response to a bot prompt (e.g. "What's a country you would like to visit?"), then the RG responds with an 'infilled' remark (discussed below) or an interesting fact (i.e. 'TILs' scraped from the /r/todayilearned subreddit) about the entity. These conversation starters serve the purpose of drawing the user into a more conversational dialog about the entity before proceeding to a more content-rich discussion of it.

Discussing the entity in depth. If the user responds positively to our initial discussion of the entity, we begin a "Discuss in depth" conversation loop (see Fig. 6). Our bot provides a summary of some section of the entity's Wikipedia article and handles the user's sentiments, opinions, and questions appropriately before checking if the user would like to continue with the discussion. If the user responds affirmatively, we suggest another section for discussion, otherwise we exit the RG. This setup ensures that the user is not overly fatigued by the amount of information generated in these section summaries, while allowing interested users to discuss engrossing topics in great depth.

A short example Wiki interaction is shown in Turns 6 through 10 of Table 1.

E.3.3 Template-Based Infilling

To provide the user with rich, coherent conversation for a wide class of entities, we developed a novel method—*infilling*—which generates interesting remarks from handwritten templates based on relevant context. For example, given the actor Keanu Reaves as the current entity, the template *I love how **[actor]** acted in **[film]**, especially their* ***<mask>*** might be infilled as follows: *I love how **[Keanu Reeves]** acted in **[The Matrix]**, especially their **ability to freeze time***. By defining a diverse set of templates for each entity category, we are able to provide expressive yet controllable conversation on many different types of entities. In effect, this acts as a more flexible version of standard slot-filling methods that does not require a structured knowledge base.

Infilling has the following steps:

- A set of templates and appropriate contexts is **retrieved**. Given some entity, we select a set of handwritten templates based on its Wikidata category (e.g. *actor*, *musical instrument*). For each template, we retrieve an appropriate short context from Wikipedia (approximately 3 sentences) using the mean-pooled GloVe-based method of (Arora et al., 2016).

- Given each (context, template) pair, an **infiller** model fills in the blanks. This is parameterized by a BART-base model trained on a dataset generated by ~ 4300 examples, mostly generated using GPT-3 (Brown et al., 2020) and augmented by hand-written examples.

- The infills are **reranked** by an aggregate DialogRPT (Gao et al., 2020) and likelihood score as measured by a GPT-2-medium model fine-tuned on Empathetic Dialogues.

E.3.4 TIL's: Conversational Paraphrasing

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 finetuning 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.

E.4 Opinion

Exchanging opinions is a core part of social chitchat. 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.

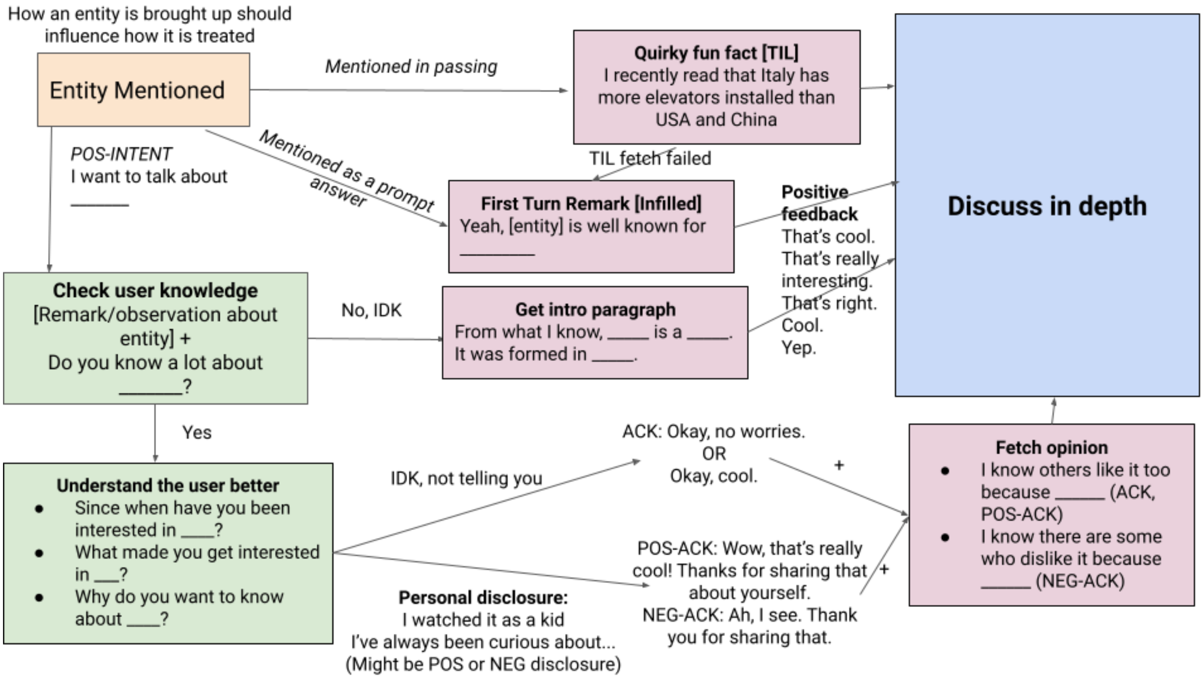

How an entity is brought up should influence how it is treated
Entity Mentioned
Mentioned in passing
Quirky fun fact [TIL]
I recently read that Italy has more elevators installed than USA and China
Discuss in depth
POS-INTENT
I want to talk about ____
Mentioned as a prompt answer
TIL fetch failed
First Turn Remark [Infilled]
Yeah, [entity] is well known for ____
Positive feedback
That's cool.
That's really interesting.
That's right.
Cool.
Yep.
Check user knowledge
[Remark/observation about entity] +
Do you know a lot about ____?
No, IDK
Get intro paragraph
From what I know, ____ is a ____.
It was formed in ____.
Yes
ACK: Okay, no worries.
OR
Okay, cool.
+
Fetch opinion
I know others like it too because ____ (ACK, POS-ACK)
I know there are some who dislike it because ____ (NEG-ACK)
Understand the user better
Since when have you been interested in ___?
What made you get interested in ___?
Why do you want to know about ___?
IDK, not telling you
POS-ACK: Wow, that's really cool! Thanks for sharing that about yourself.
NEG-ACK: Ah, I see. Thank you for sharing that.
+
Personal disclosure:
I watched it as a kid
I've always been curious about...
(Might be POS or NEG disclosure)

Figure 5: The Wiki RG conversational flow: possible user responses are captured in the edge labels, while bot responses are represented by the vertices.

E.4.1 Data

To collect both positive and negative opinions, we queried a Twitter stream[25] 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 on 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.

E.4.2 Behavior

Currently, the Opinion RG activates when the user mentions one of the whitelisted entities (e.g. Table 1, Turn 8). We ask whether the user likes the entity and classify their response using the CoreNLP sentiment classifier (Section C.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.

E.4.3 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:

- (i) `ALWAYS_AGREE` – 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

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

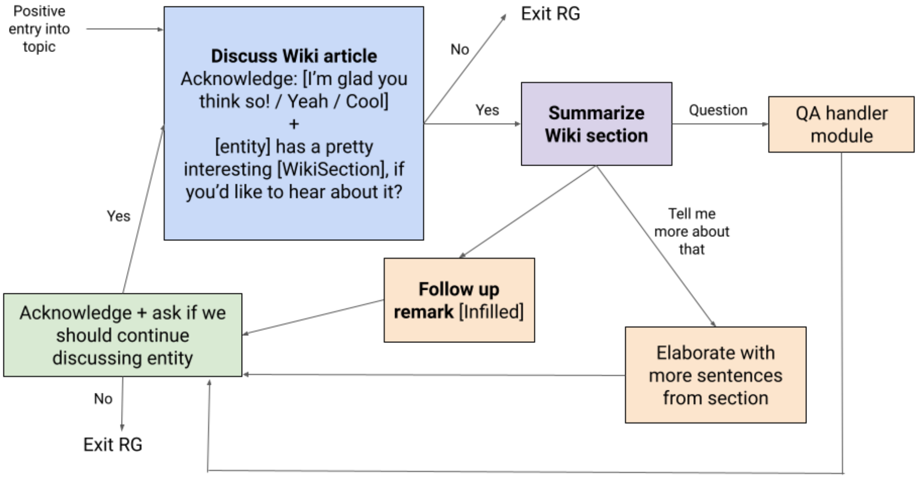

Positive entry into topic
Discuss Wiki article
Acknowledge: [I'm glad you think so! / Yeah / Cool]
+
[entity] has a pretty interesting [WikiSection], if you'd like to hear about it?
No
Exit RG
Yes
Summarize Wiki section
Question
QA handler module
Tell me more about that
Yes
Follow up remark [Infilled]
Acknowledge + ask if we should continue discussing entity
Elaborate with more sentences from section
No
Exit RG

Figure 6: The Wiki RG "Discuss in depth" conversational loop

Policy Name	Continuation Rate (95% CI)
CONVINCED_AGREE	.527 ± .0349
ALWAYS_AGREE	.587 ± .0086
LISTEN_FIRST_DISAGREE	.587 ± .0128

Table 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).

- (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 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.

E.5 Food

The Food RG also focuses on scripted responses to discuss foods and give suggestions. It is often activated at the beginning of the conversation when Neural Chat RG prompts a user for what they have eaten today. The Food RG then goes through a sequence where it asks the user about their favorite variant of that food (e.g. favorite pizza topping), mentions the bot's favorite variant, and possibly provides a fun fact about the food. The Food RG is backed by food data scraped from Wikipedia structured in such a way that subclasses and variants of food are linked to each other. It also uses templated responses with neural infilling to generate descriptions of foods or comments on what the user likes, allowing for variation and flexibility for more interesting responses.

E.6 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.[26] 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 1). The RG uses treelets (Section 2) to organize the dialogue graph, hand-written templates to form the bot utterances, and a mixture of regexes and the CoreNLP sentiment classifier (Section C.1) to classify the user's responses.

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

E.7 Music

Similar to the Movies RG, the Music RG is designed to deliver scripted conversations about musical entities that the user specify. 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 of the weaknesses of scripted conversations mentioned in Section E.6.

E.8 Sports

The Sports RG is designed to deliver up-to-date and high-quality conversations on a sport for which the user expresses interest. Currently, we support conversations on NFL football and NBA basketball, the two most-watched sports in the US. When prompted to discuss sports, the user is asked if they are a fan of these two sports. If so, they are asked for their favorite team, but otherwise the conversation moves to a different RG. The RG supports detailed, factual conversation on the user's favorite team, as well as their favorite player on that team. The Sports RG is backed by an ESPN API scraper that pulls information on all NFL and NBA teams (their game schedule, their roster, wins/losses, game analysis, etc.) and facts about all players (their age, position, college, statistics, and expert analysis on their overall play). For example, if the user is a fan of the Denver Broncos, the RG is capable of discussing the Broncos' most recent game (who won/lost, what the score was, what player played well, etc.) and then transitions into discussing a specific Broncos player from the game that the user likes. By utilizing automatic summarization, we are able to intersperse current, specific analysis of their favorite player or team that comes directly from ESPN analysts, giving the conversation a sophisticated and natural tone.

E.9 Fallback

When all other RG's fail to produce a suitable response, we rely upon two fallback RG's that always execute. The Neural Fallback RG selects a fallback responses generated by the DistillBlender model (Section 3.1), with all questions removed. Given that the model is trained on end-to-end dialogue, we find that this is a good conversational baseline. If the Neural Fallback RG fails, we resort to the Fallback RG, which returns a pre-written generic fallback (e.g. "Sorry, I don't know how to answer that").

F High-Initiative User Handling: Example Dialog Flows

Our system uses high-initiative handler functions to detect 18 distinct types of one-turn digressions that interrupt the conversational flow. General examples include complaints and clarifying questions. A comprehensive list of high-initiative situations that our bot can handle and corresponding examples of bot responses are provided in Table 6. These functions can be started in any RG, and allow each RG to pause the flow of the current dialogue tree to handle high-initiative behaviors before either returning to the original conversation or changing the subject.

To avoid repetitiveness, high-initiative handlers are generally non-reentrant, and cannot be triggered consecutively.

G Additional Neural Modeling Experiments

In addition to BlenderBot-3B, we experimented with a variety of autoregressive and non-autoregressive models for text generation. All models are evaluated on an internal dataset of 517 conversation excerpts from early 2021 where the Neural Chat RG was active **AP2021**. We perform qualitative evaluation by passing in each conversational excerpt to the model of interest, and comparing the resultant generation(s) with the original neural generation from GPT2ED. We detail results here.

DialoGPT. DialoGPT (Zhang et al., 2020b) is a GPT-Medium model that has been further fine-tuned on a set of Reddit threads serving as conversational corpora. We evaluated this model offline on a set of excerpts in two settings: 1) **zero-shot** and 2) with fine-tuning on Empathetic Dialogues **(DialoGPT2ED)**. In the **zero-shot** setting, the bot re-

High-Initiative User Behavior	**User utterance**	**Possible response**
Was misheard	That's not what I said.	Sorry for the misunderstanding; can you repeat that?
Complains bot was unclear	I don't know what you are talking about.	Oops, let me say that again.
Complains bot was repetitive	Alexa, you already told me that.	You're right, I forgot. Sorry. Let's move on to something else. [NEW_TOPIC]
Complaints about privacy	That's none of your business.	Sorry, maybe that was too personal. I'm happy to talk about something else. [NEW_TOPIC]
Asked about weather	Do you like the weather?	I live in the cloud so I'm not sure what the weather is like on earth!
Said Alexa command	Alexa, play Despacito!	This is an Alexa Prize Socialbot... I can't execute typical Alexa commands...But I'd love to keep talking to you! As I was saying, [PREV]
Wants name correction	Actually, that's not my name.	Oops, it sounds like I got your name wrong. I'm so sorry about that! I won't make that mistake again.
Requested repetition	Alexa, can you say that again?	Oops. Let me repeat that. [PREV]
Requested own name	What's my name?	If I remember correctly, your name is [NAME]
Got cut off	But—	Oops, I think my microphone stopped working for a second. Can you say that again?
Asked for our age	How old are you?	Hmm, I don't know. I'm just a bot, and I don't think we have birthdays.
Asked clarifying question	Wait, did you say "elephants?"	Yes, that's what I said.
Questioned bot's humanity	How can you listen to music if you're a bot?	Well, I'm just a bot, but I live vicariously through others.
Asked personal question	What do you like?	Thanks for asking! [NEURAL]
Interrupted with question	Wait, I have a question.	Sure, what's up?
Expresses desire to chat	Let's chat!	Okay. What would you like to talk about?
Asked for story	Tell me a story.	Here's a story that someone once told me. [STORY]
Gave bot compliment	I like talking to you!	Thanks for saying that. I'm still learning, but it means a lot to hear that from you.

Table 6: Excerpts of possible bot responses to 18 high-initiative user behavior modes. [PREV] refers to the previous bot utterance; [NEW_TOPIC] refers to a sampled prompt from a new RG. [NAME] is the user's name as obtained in the opening turns, and [NEURAL] refers to a DistillBlender-based random response. [STORY] is a handwritten anecdote, omitted here for brevity.

sponds 18% of the time with dirty jokes or memetic content unsafe for open-domain conversation on **AP2021**. After fine-tuning, **(DialoGPT2ED)** responds almost identically to GPT2ED on **AP2021**: qualitatively, the lift from DialoGPT2ED is essentially zero. Hence, this system was not deployed.

DistillBART. DistillBART is our in-house distilled version of BART (Lewis et al., 2020), a model consisting of a non-autoregressive encoder and an autoregressive decoder, each with 12 layers. Notably, this model has decoding complexity $\mathcal{O}(EN + DN^2)$, where N is the sequence length, and E, D are the sizes of the encoder and decoder stacks, respectively. Following results by (Kasai et al., 2020) in the domain of neural machine translation, we hypothesized that we could decrease latency while improving performance by decreasing D; i.e. removing decoder layers and training the decoder via distillation. We performed DistillBERT-style distillation, distilling a BART-Large fine-tuned on Empathetic Dialogues (BARTED) into versions with 6 **(DistillBART-6)** and 3 **(DistillBART-3)** decoder layers. Weight initialization followed a previous setup for BART distillation (Shleifer and Rush, 2020). As baselines, we also trained equivalently-sized models without distillation.

In practice, BART suffered from 1) high latency and 2) mediocre response quality. BART was unable to generate coherent responses stochastically, necessitating the usage of beam search, which hurt decoding speed. On **AP2021**, average decoding speeds for the 12, 6, and 3 layer models were 894ms, 998ms, and 895ms, showing no significant latency gains, which is attributable to the quadratic dependence within the decoding computation on sequence length; i.e. $N^2 \gg D, E$. Furthermore, while distillation certainly resulted in qualitatively better generations on **AP2021** than those of non-distilled models, as shown in Table, there was a sharp dropoff in generation quality on all models except the full-sized BARTED teacher. As BARTED was the only usable model, and yielded generations qualitatively similar to GPT2ED, we did not deploy this system.