Advancing Open Domain Dialog: The Fifth Alexa Prize SocialBot Grand Challenge

Michael Johnston **Cris Flagg** **Anna Gottardi** **Sattvik Sahai** **Yao Lu** **Lucy Hu**

Samyuth Sagi **Luke Dai** **Prasoon Goyal** **Behnam Hedayatnia** **Di Jin**

Patrick Lange **Shaohua Liu** **Sijia Liu** **Daniel Pressel** **Hangjie Shi** **Zhejia Yang**

Chao Zhang **Desheng Zhang** **Leslie Ball** **Kate Bland** **Shui Hu** **Osman Ipek**

James Jeun **Heather Rocker** **Lavina Vaz** **Akshaya Iyengar** **Yang Liu**

Arindam Mandal **Dilek Hakkani-Tur** **Reza Ghanadan**

Abstract

Creating conversational dialog systems that are able to converse naturally and engagingly with humans on any topic remains one of the fundamental challenges of artificial intelligence. The Alexa Prize SocialBot Grand Challenge was launched in 2016 to take on the problem of enabling conversational systems to support natural, sustained, coherent, and compelling open-domain dialog. The competition enables university teams from around the world to test their innovations at scale with Alexa users. The 5th SocialBot Grand Challenge (SGC5) expanded the competition to include both a live judged competition on system performance and a Science and Innovation prize to acknowledge underlying scientific achievements. SGC5 also added multimodality to the challenge and encouraged teams to augment their open-domain conversations with multimedia content and multimodal interaction. The challenge included an extensively updated version of the CoBot (Conversational Bot) Toolkit, along with numerous models and APIs, including topic and intent classifiers, offensive content classifiers, pre-trained neural response generators and rankers, and multimodal support so that teams could land running and focus on building compelling multimodal conversational experiences. Use of large language models (LLMs) was a key theme in the fifth iteration of the competition and, in addition to neural response generators fine-tuned on previous Alexa Prize conversational data, we provided APIs and fine-tuning capabilities enabling teams to make use of the 20 billion parameter Alexa Teacher Model LLM. The paper describes the operation of the competition and capabilities provided to teams. We outline and summarize the advances developed both by university teams and the Alexa Prize team in pursuit of the Grand Challenge objective, including use of LLMs and instruction prompting for dialog control, synthetic data and knowledge generation, multimedia response generation, and dialog evaluation.

5th Proceedings of Alexa Prize SocialBot (Alexa Prize 2023).

1 Introduction

The Alexa Prize[1] is an Amazon Alexa sponsored program that has enabled hundreds of university students and faculty to compete in advancing the state-of-the-art in conversational AI. Since 2016, the SocialBot Grand Challenge has hosted a competition among universities across the world to compete in creating the best *SocialBot*, i.e., an Alexa skill that can engage in extended open-domain dialogs with users around popular topics and current events [21]. Since 2021, the TaskBot Challenge has engaged teams with building conversational assistants that can assist users in completing complex tasks such as recipes or DIY projects [16]. The SimBot Challenge, started in 2022, pushes the boundaries of embodied conversational AI by challenging teams to build SimBots that users can instruct to complete tasks in a simulated 3D environment ([46]). One of the key advantages of the program is that it enables university teams to rapidly test and iterate on their approaches through testing with real world users at scale through Alexa.

Conversational AI remains one of the most challenging problems in artificial intelligence. Compared to tasks like speech recognition, text classification, or image recognition, the creation of interactive conversational dialog systems requires high performance and seamless integration of numerous diverse capabilities including natural language understanding, natural language generation, information retrieval and search, and dialog management yielding an artificial conversation partner that can engage effectively with a human user. While significant strides forward have been made in task-oriented dialog for voice assistants the ability to support engaging social chat on any topic has remained more elusive. While contemporary large language models demonstrate ability to respond to open-domain requests and have the ability to handle follow on questions, they are text-based and not well suited to the affordances of voice interaction and more importantly do not actively take initiative to draw out the user and keep them engaged in conversation. To address this challenge and drive towards conversational AI systems that can communicate like a human participant at a cocktail party or a trusted friend, Amazon launched the Alexa Prize SocialBot Grand Challenge (SGC) in 2016. SGC was the inaugural Alexa Prize competition and challenged teams from around the world to build agents that can converse coherently and engagingly with humans for 20 minutes or more, and obtain a 4 out of 5 rating from users interacting with them.

Figure 1: SocialBot on Multimodal Devices

Users launched the Alexa Prize SocialBot Grand Challenge 5 by saying “Alexa, open Alexa Prize”. After holding a conversation, users were asked to rate the conversation on a scale from 1 to 5 stars and provide free-form feedback. The ratings and feedback were both shared back with the university teams to help them improve their SocialBots.

The SocialBot challenge has evolved over the years, and there were significant changes and advances to the competition for SGC5, including the introduction of a distinct prize for Scientific Invention and Innovation, the addition of multimodal interaction to the SocialBot challenge, and the addition of support for incorporation of large language models that teams could apply in various ways in building their SocialBots.

In previous iterations of the Alexa Prize SocialBot Grand Challenge, there was a single Prize track where teams competed to achieve the highest satisfaction ratings in a judged finals event. For the first time we introduce a second award, the Scientific Invention and Innovation Prize. Teams compete to showcase their advancement of the science of open-domain conversations through submission of short technical innovation papers and detailed final presentations of their technical approach. With the traditional Prize we capture how the teams are advancing the user experience. In the Science

[1] `https://www.amazon.science/alexa-prize`

prize we evaluate what is under hood; that is, how teams are innovating to advance the science of open-domain conversational AI.

While the origins of conversational assistants are very much as a voice-first experience, devices which can support a multimodal combination of visual presentation and touch input combined with spoken input and output are increasingly commonplace. In recognition of the prevalence of multimodal interaction, this year is the first SocialBot Grand Challenge to include support for multimodal interaction as part of open-domain dialog. As detailed in Section 2.1.3 Multimodal Experience, we extended the tools we provide to teams to allow for visual presentation. Teams used these capabilities to provide on-screen text, relevant visual images, avatars, and background videos. The teams also used the visual interface to provide multiple choice question answering and feedback. Figure 1 shows an example of a multimodal display on an Echo Show 8 device from our sample SocialBot.

Large language models (LLMs) have played a significant role in the SocialBot Grand Challenge since early in the challenge, but nothing compared to their front stage role in SGC5. Since SGC3 in 2019, Alexa Prize has provided teams with neural response generation model APIs based on GPT-2[41] and fine-tuned on previous Alexa Prize conversations. Some teams used this model as a fallback when other responders did not provide a high quality response. Others such as Proto [44], the second place winner in SGC4, used GPT-2 based models alongside fine-tuned versions of BlenderBot [47] and deterministic responders, driving conversation using a 'neuro-symbolic cocktail' and employing a response selection component to triage among responses from different models. High capacity LLMs with the ability to take explicit human language instructions in their prompt ([39]), ability to perform in-context learning ([3]), and emergent abilities to handle unexpected tasks have rapidly come to dominate work in natural language processing and dialog. They have also captured the imagination of both enterprises and the general public and brought the concept of open-domain dialog systems out of the lab and into public discourse. Perhaps not surprisingly, these trends have influenced the SGC5 teams building their SocialBots. Rather than simply using causal language models as response generators based on context, teams are exploring using LLMs to control the dialog layer and making extensive use of instruction prompting to control LLM response generation. Teams are also using LLMs offline to drive synthetic data generation and using in-context learning to perform tasks such as intent classification and automated dialog evaluation. To support this work, in SGC5 we introduced model APIs and fine-tuning capabilities that support various versions of the Alexa Teacher Model (AlexaTM) [49]. We provided both 5B and 20B sizes and versions with pre-training specifically targeting conversational applications. In addition to AlexaTM, teams made use of numerous open source LLMs including Vicuna 13B [5], Falcon 40B [2], BlenderBot 3 [47], FLAN-T5 ([6]) and others. It is important to note that the affordances of spoken dialog interaction are quite different from text chat and teams had to make significant innovations in order to reduce latency and optimize the user experience. These and other aspects of LLMs in SGC5 are discussed in more detail in 3 Scientific Advancements.

The fifth SocialBot competition began with a comprehensive three-day Bootcamp in December 2022. During this event, nine university teams were invited to receive Amazon Web Services (AWS) training, CoBot (Conversational Bot) tooling, and hands-on development experience. Throughout the Bootcamp, all nine teams successfully developed a bot using a baseline model provided by the Alexa Prize team, utilizing the resources offered by AWS and Alexa. Following this training, teams put their efforts to refine and enhance their bots until the end of January 2023, ultimately completing the skill certification process required for deployment with Alexa users.

In the context of open domain dialogs, the SocialBots need to be able to respond to any type of conversation. When presented with topics that are sensitive, controversial, or inappropriate the SocialBots must responsibly filter their responses to ensure the SocialBot does not inadvertently engage or promote these topics. The certification process tested the SocialBots to ensure they were able to engage users in a responsible manner and appropriately filter and respond to inappropriate and sensitive topics. Once the SocialBots passed certification, internal Amazon beta testers provided initial feedback to the teams.

All nine teams progressed from the launch phase and advanced to the Semifinals from May 8, 2023 through June 23, 2023. From the Semifinals, five teams successfully advanced to the Finals phase and ultimately competed for top honors in the closed door Finals event on August 2, 2023.

For the Scientific Invention and Innovation Prize, teams focused on advancing the state-of-the art in open domain conversation. Teams submitted proposals describing their innovations and six teams

were chosen present their research to a panel of judges from Amazon Alexa with three teams selected on August 29, 2023 as the final winners.

In Section 2 Capabilities Provided to Teams we provide details on the capabilities provided to the teams. Section 3 Scientific Advancements summarizes the scientific advancements in the challenge both for participant teams and from the Alexa Prize team. The evaluation and performance of the SocialBots is detailed in Section 4 SocialBot Performance: Discussion, and overall insights gathered from the SocialBot Challenge and concluding remarks are in Section 5 Conclusion.

2 Capabilities Provided to Teams

To facilitate research on SocialBot and advancing the science of multi-modal conversational AI, the university teams were granted exclusive access to a range of Amazon Alexa resources, technologies, and experts in the science and engineering of Conversational AI systems. The following is an overview of the resources that were made available.

2.1 Conversational Bot Toolkit (CoBot)

We provided the SocialBot teams with CoBot [23], a conversational bot toolkit in Python for natural language understanding and dialog management that has been used across both SocialBot and TaskBot competition tracks since 2018. CoBot includes a set of tools, libraries, and base models to help develop and deploy open-domain or multi-domain conversational experiences through the Alexa Skills Kit [28] and Amazon AWS (see figure 2). CoBot's modular, extensible, and scalable design was developed based on learnings from previous Alexa Prize competitions and provides abstractions that enable the teams to focus more on scientific advances and reduce time invested into infrastructure, hosting, and scaling.

For the fifth SocialBot Grand Challenge, we overhauled CoBot's deployment infrastructure to address major pain points in the continuous integration and continuous delivery (CI/CD) pipelines. CoBot is designed to give teams a CI/CD environment with which incremental code changes on bots are merged, delivered and released frequently and reliably to testing and production environments. This year, we minimized the iteration cycle time for each deployment by decoupling component deployments into separate pipelines that could be deployed independently, reducing the deployment time from 20 minutes to under 5 minutes. We estimate this optimization has saved 195 days worth of development time throughout SGC5, enabling students to invest that time back into their research efforts. In addition, we also made significant changes in CoBot to support hosting large language models (LLMs), as much as 640 GB, which is 160 times larger than previously hosted in CoBot.

To support multimodal SocialBot development for the first time, we built on the initial set of Alexa Presentation Language (APL) [1] templates provided in CoBot for the TaskBot competition to incorporate five new templates custom-designed for SocialBot, as described in Section 2.1.3 Multimodal Experience.

Lastly, we provided teams with a sample bot, Section 2.1.5 Sample Bot, implemented using CoBot to illustrate simple examples of how to use all of CoBot's provided APIs in the context of a basic SocialBot. For SGC5 we expanded the set of API offerings to include our latest generative neural models, a News retrieval API, and APIs for response ranking, contradiction detection, and utterance rewriting, as described in Section 2.1.4CoBot APIs. Figure 6 illustrates the CoBot architecture used in the sample bot. The modular design accelerates experimentation by enabling researchers to test and evaluate innovations in different components.

We continue to invest in CoBot as our flagship toolkit for building open-domain dialog systems and constantly extend CoBot's capabilities to support the evolving demands of SocialBot, TaskBot, and other potential conversational bots in the future.

2.1.1 CoBot Deployment Infrastructure

CoBot is designed to give teams a CI/CD environment with which to rapidly deploy and iterate on bots. Figure 2 captures the system deployment architecture for a standard CoBot-based bot. CoBot uses AWS Lambda as a serverless infrastructure to host local modules. The Lambda is the endpoint of an Alexa skill and receives requests from the skill. CoBot uses Amazon Elastic Container Service

(Amazon ECS) and Docker to deploy and host bigger and long-running remote Docker modules. The module services sit behind Amazon Application Load Balancers (ALB). CoBot points the Lambda to the ALBs so that the bot can send requests from Lambda to remote Docker modules.

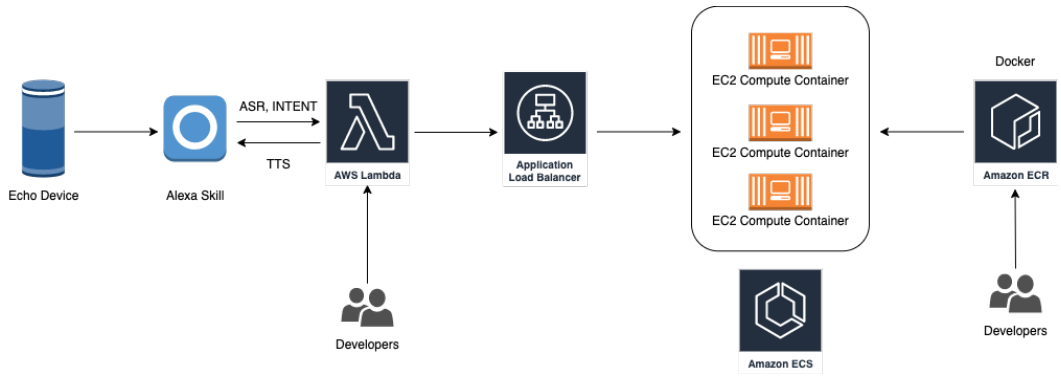

Figure 2: CoBot System Architecture

CoBot builds a continuous delivery pipeline for an AWS Lambda application with AWS CodePipeline. The Lambda CodePipeline will monitor the Lambda CodeCommit repository for new commits, build the Lambda, and deploy it with AWS CloudFormation.

New for SGC5, CoBot now builds a separate AWS CodePipeline for each remote Docker module. The CodePipeline monitors a module's CodeCommit repository for new commits, uses AWS CodeBuild to create a Docker container image and push it into Amazon Elastic Container Registry (Amazon ECR), and uses AWS CloudFormation to deploy the container image to production on Amazon ECS. A Flask [17] application sits behind an Amazon ALB to provide the scalability and resiliency to handle traffic from Alexa Prize users. Figure 3 shows the new design for deployment pipelines for remote Docker modules and Lambda.

The decoupling of remote Docker module pipelines allows researchers to only deploy the changes they need, without having to redeploy all modules. In addition, it facilitates tailoring of the GPU instance type used for hosting GPU-based containers to be different for each remote Docker module. This is a major cost-saving win, since researchers no longer need to use the largest GPU instance type for every GPU-based remote Docker module.

2.1.2 Large Language Model Support in CoBot

Hosting large language models poses some unique challenges. Due to the large size of these models, they generally have slower inference speeds and take a long time to load into memory. These restrictions make it difficult to host a real-time, low-latency service that can scale up to serve high volumes of user traffic. We added special provisions in CoBot to facilitate hosting LLMs as remote Docker modules (as described in section 2.1.1).

Inference speed: We utilized a lightweight Flask-based server to host LLMs in CoBot, and the inference engine was built using the HuggingFace transformers library [54]. All models were converted to bfloat16 for inferencing. This reduced their memory footprint, resulting in faster inference per GPU, and also enabled the model to be split between fewer GPUs, providing further gains in latency. We also experimented with numerous EC2 types and determined that G5 instances[2] with NVIDIA A10 GPUs[3] provided the fastest inference speeds with relatively low costs. We shared an example implementation of hosting a 5 billion parameter version of Alexa Teacher Model [49] to help university teams incorporate all these optimizations.

Autoscaling: CoBot remote Docker modules provide built-in auto-scaling policies based on CPU and Memory utilization. However, while hosting LLMs these policies proved ineffective for the following two reasons:

[2] https://aws.amazon.com/ec2/instance-types/g5/

[3] https://www.nvidia.com/en-us/data-center/products/a10-gpu/

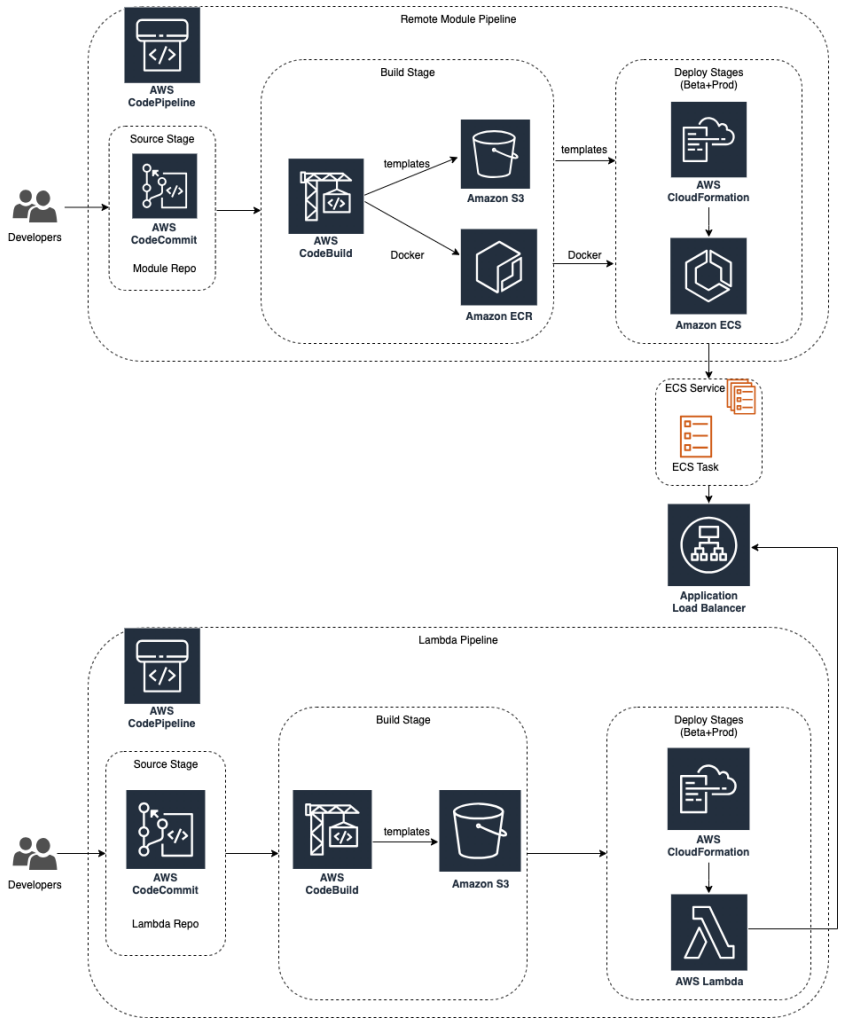

Figure 3: CoBot Deployment Pipelines

1. While CPU and memory utilization are usually the right bottlenecks to monitor and hence serve as reasonable metrics for scaling services, GPU memory/utilization is usually the bottleneck for LLM inference. In our experiments, we observed that CPU and memory utilization were not strongly correlated with the number of invocations to the service or the number of LLM inferences in progress. Also, because of the large model sizes we have to pre-load the model in GPU memory instead of loading it for each incoming request. This also makes it infeasible to use GPU memory utilization as a metric for scaling. We experimented with GPU utilization as the scaling metric and found it to be better correlated with the overall invocation volume, but it was strongly dependent on model architecture. To get around all these limitations, we decided to create a scaling metric based on the invocation count, current task count, and historic model latency. The new metric is defined as follows:

$$Task\_Utilization = \frac{latency * request\_count}{task\_count * 60} * 100\% \quad (1)$$

Where, latency is the historical p90 latency of the service (averaged over a minute) in seconds, request_count is the number of incoming requests per minute, and task_count is the number of ECS tasks running.

2. As we use Amazon ECS[4], scaling of services is done based on tasks which contain a Docker container specific to the application. Scaling up adds more tasks and scaling down

[4] `https://aws.amazon.com/ecs/`

removes running tasks from the ECS cluster. For LLM-based services, adding a new task was much slower than most services as it requires copying over all the model weights. We experimented with multiple approaches for this, including building a Docker image with the model parameters in it and loading model weights from S3. All these methods were prohibitively slow, taking in the order of hours to copy models into a new task. Due to this, auto-scaling was too slow to be able to keep up with user traffic, and we had to maintain a large buffer of extra tasks to ensure uninterrupted user performance. As a solution to this, we introduced Amazon Elastic File System (EFS)[5] based model loading. In this approach the model weights are copied into EFS Volumes during the build stage of the deployment pipeline. These volumes are then just attached to all ECS tasks which can read from it like a typical file storage. This approach offloaded the slow model loading process to the build stage and reduced the new task spin up time from hours to minutes.

Deployment pipeline: Another goal for our team was to facilitate fast experimentation for teams. For hosting large language models, we added a step in the build stage where model artifacts are copied from Amazon S3 to EFS. Due to the large size of these models, the time duration of this copy operation can be in the order of hours. To optimize this process, we built a checksum mechanism to provide low amortized build time. This was achieved by computing a checksum based on all model artifacts. This checksum is stored in a DynamoDB table and is checked at every build request. CoBot only copies the model from S3 to EFS if the checksum is different from the one in DynamoDB. This way we avoid unnecessary copying of data from S3 to EFS and in turn, significantly reduce the average build time, providing a better developer experience.

Multi-region support: Finally, we added multi-region remote Docker modules in CoBot. This was done to empower teams to select the best regions for each remote Docker module based on instance availability. GPU based instances are in high demand and can sometimes cause deployments to get stalled due to unavailability of new instances. In the worst case, this could cause negative user impact.

2.1.3 Multimodal Experience

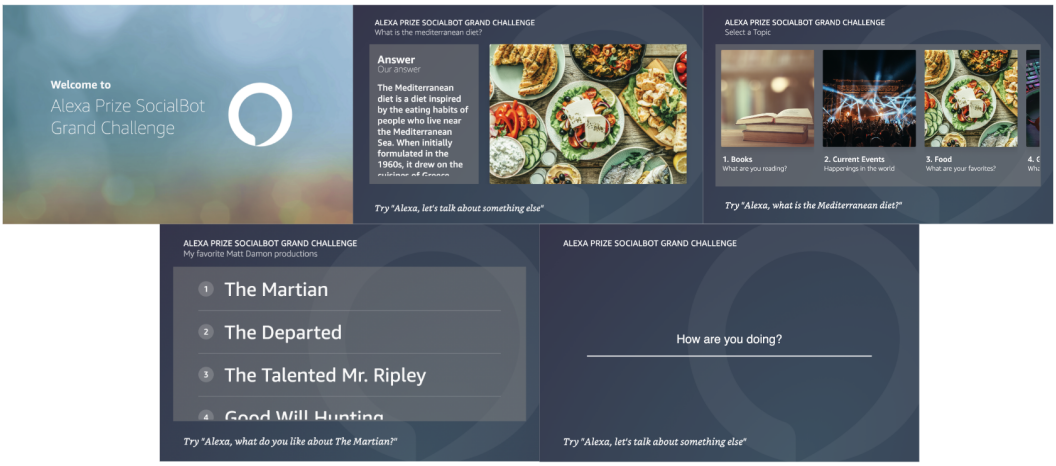

Figure 4: Default SocialBot Templates

To enable developers to build interactive voice and visual experiences, the Alexa Skills Kit provides a visual design framework called Alexa Presentation Language (APL) [1]. APL includes visual elements that scale across device types and can support both voice and touch interactions. This year we introduced APL capabilities to SocialBot teams for the first time. Custom APL templates were distributed in the CoBot Toolkit along with commands that could be executed on these templates. The templates include a Chat Template, Detail Template, Image List Template, Landing Template, and Text List Template (see Figure 4). These were provided in the form of Python objects with methods

[5] `https://aws.amazon.com/efs/`

to manipulate components and render the final JSON documents from which an Alexa device with a screen generates its display.

Methods to manipulate templates were robust, with each template having dozens of options that could be configured. Team NAM (Stevens Institute of Technology) [33], for example, was able to use a looping video as a background in addition to displaying text and images. This was done using an option in CoBot that allowed for the addition of a background image or video source. Beyond using the methods native to CoBot, teams could also use the JSON document for the sample templates to further customize their experience by adding components or tweaking existing components.

Along with the templates, we also provided APL commands that enabled users to scroll and have Alexa speak text written on the screen. Additionally, Team Alquist (Czech Technical University) [25] added karaoke functionality, or the ability to highlight lines or blocks on screen as Alexa is reading out those lines or blocks. With their permission, this feature was subsequently added to our toolkit for use by other teams. In addition to the commands, we enabled options for multi-turn touch interactions with the APL display, where the user requests, voice or touch based, were sent in place of a user utterance for processing by the Lambda.

Besides newly developed SocialBot features, templates and commands that were developed for the purpose of the Alexa Prize TaskBot Challenge were also made available for the SocialBot track. These included three standard APL templates available to all Alexa skill developers (Alexa Text List, Alexa Image List, and Alexa Detail) and a custom video template with a clickable media player.

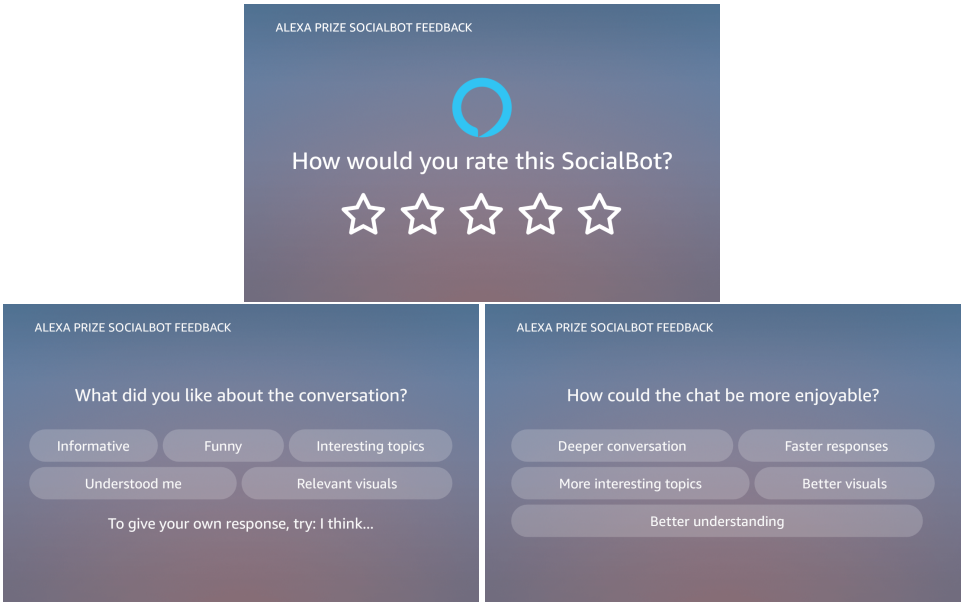

Figure 5: SocialBot Multimodal Feedback Screens

2.1.3.1 Multimodal Feedback Flow

It is important for researchers to get direct real-world feedback from Alexa users to evaluate and improve their innovations. Real world traffic differs significantly from curated datasets. Providing multimodal feedback makes it faster and easier for users to comment on their conversations and increases the amount of data that teams receive about their interactions.

Before SGC5, a voice feedback flow was already implemented to get user feedback after each conversation. After the end of the SGC5 Semifinals Interaction Period, we introduced a multimodal feedback flow to accompany the voice feedback flow that users go through at the conclusion of each SocialBot conversation (Figure 5). Providing a GUI enables users to more quickly and easily rate the bots without waiting for the prompt to complete.

The first screen uses a truncated ratings question that allows users to enter a rating from one to five. Next, we replaced the free-form feedback question with a more targeted set of questions. For conversations that a user rated highly (4 or 5), they are asked, "What did you like about the conversation?" Five options are provided that may be toggled through touch interaction: Informative, Funny, Interesting topics, Understood me, and Relevant visuals.

In contrast, users that gave the conversation a lower rating are asked, "How could the chat be more enjoyable?" Users may select from the following options: Deeper conversation, Faster responses, More interesting topics, Better visuals, and Better understanding.

The GUI-based rating and feedback functionalities on multimodal devices offer substantial improvement as evidenced by the following observations. Users have expressed their inclination to provide ratings and feedback through touch interactions. Subsequent to the activation of GUI cards, 45% of the conversations with ratings or feedback incorporated the GUI features. The integration of GUI elements also led to an increased rate of user-provided ratings, increasing from 45% to 55% of all conversations lasting over 30 seconds. In addition, users display a willingness to give feedback using multi-choice options over free-form responses. In the initial week after the introduction of GUI features, approximately 40% of all feedback on multimodal devices leveraged the touch-based functionality. Finally, there is also a notable enhancement in the percentage of responses that included feedback (as opposed to just giving a rating), increasing from 51% to 63%.

2.1.4 CoBot APIs

For this fifth iteration of the competition, we expanded the set of API offerings that participating teams received. These new APIs spanned from generative neural models to classifiers, rankers and a news retrieval API. This section contains some details about these new APIs.

2.1.4.1 Alexa Teacher Model LLM APIs

We provided the teams with APIs for invoking the Alexa Teacher Model [49]. These models are sequence-to-sequence transformer models hosted in ECS clusters. More specifically, the following three flavors of AlexaTM were provided as endpoints accessible to the teams:

- Vanilla AlexaTM 20B - This is a 20 billion parameter model trained on a multilingual dataset as described by Soltan et al. [49]
- Conversationally pre-trained AlexaTM 20B - This is a version of the 20 billion parameter model built on the vanilla AlexaTM model and further trained on conversational data.
- Conversationally pre-trained AlexaTM 5B - This is a version of the 5 billion parameter model built on the vanilla AlexaTM 5B model and further trained on conversational data.

2.1.4.2 Neural Generators

For SGC5 we provided three new neural response generators (NRG): the empathy NRG, topic NRG, and NRG XL. These models are based on the GPT-2 [41] architecture. The empathy NRG is trained to generate text conditioned on dialog context and a conditional attribute that takes a value of *condolence*. This model can be used to generate empathetic responses. The topic NRG also has a similar interface, with the only difference being that the conditional attribute can be used to pass the topic for which the response should be generated. Finally, we also added a NRG XL model based on the GPT 2 XL model. This model was provided through the NRG API introduced during SGC3 [14].

2.1.4.3 News Retrieval API

We provided an API that facilitates the retrieval of the most recent and relevant news content based on specified keywords. The API provides various types of information, including headline, publisher details, and publication date. This involved utilization of an API Gateway to receive incoming requests, and efficiently routing them to an AWS Lambda function. This Lambda function serves as a proxy, seamlessly communicating with the service responsible for content retrieval.

2.1.4.4 Offensive Classifier

We provided an updated offensive classifier to enable researchers to detect inappropriate utterances. The model performs two functions: 1) Classify utterances between *offensive* and *not offensive*. 2) Classify utterances between *contains PII* and *does not contain PII*.

The first function of the offensive classifier API is powered by a Hybrid model which is a combination of a Roberta [34] classifier and a keyword-based classifier. The second function is implemented using another keyword-based classifier capable of flagging Personally Identifiable Information (PII) in a given utterance.

To accommodate both outputs, the API's response format contains two distinct fields for each function. The generic nature of the API allows it to be used at various stages in a SocialBot pipeline. The most common uses are 1) Filtering user utterances to identify sensitive or offensive content and invoke the relevant strategies, and 2) Filtering the outputs from a model to ensure that the bot response is appropriate.

2.1.4.5 Contradiction Detection

For SGC5, teams were provided with a contradiction detection API. This API was powered by a classifier which would determine if a generated response contradicts previous responses in the conversation. This API accepts a dialog history field along with a list of candidate responses and would generate a confidence score for each candidate response. The classifier used here was based on a Roberta-Large model trained for the task as proposed in Jin et al.[22] and described in section 3.2.2 Contradiction Detection.

2.1.4.6 Utterance Rewriting

SGC5 teams also received a new utterance rewriting API. This API uses conversation context and generates a rewrite of a given user utterance with resolved co-reference and ellipsis. The API accepts a list of utterances and is powered by a Seq2Seq model. More details are provided in the section 3.2.3 Utterance Rewriting

2.1.4.7 Response Ranking

Creating an open-domain dialog system involves building one or more response generators (RGs) that can use generative, retrieval, or template-based methods to produce candidate responses. To choose the best response, a response ranker may be used to rank the outputs from different RGs. Response rankers can be rule-based or model-based systems. Rule-based systems use manually-designed logic to rank hypotheses, while model-based approaches use machine learning models, typically neural networks, to learn how to rank candidates. As the number of RGs increases, rule-based systems become difficult to maintain, whereas model-based methods simplify the ranking process and improve performance. To allow teams to be able to scale the number of RGs included in their dialog system, we implemented a BERT-based ranker, as described in 3.2.1 Response Ranking, that takes multiple candidate responses from the response generators and assigns a numerical ranking based on the likeliness the response will produce a positive user rating.

2.1.4.8 Open-Domain Evaluation

This year we provided an additional API to assist in evaluation of open-domain social dialogs. This API takes user utterances as input and classifies them into a series of categories which capture user engagement, satisfaction, users calling out contradictions or misunderstandings among other categories. For details of the ODES (Open Dialog Evaluation Signals) dataset and categories see [29]. Underlyingly, this API uses a fine-tuned Roberta classifier [34] trained on the ODES dataset. The output is provided in the form of softmax values for 14 ODES classes. More details about ODES are provided in section 3.2.5.

2.1.5 Sample Bot

New for SGC5, we provided teams with a simplified sample bot that extended and implemented CoBot components to demonstrate the use of all the APIs and multimodal features made available in this year's competition. Figure 6 illustrates the CoBot architecture used in the sample bot. For Natural Language Processing, CoBot provides a flexible NLP Pipeline that allows developers to configure NLP modules to run in parallel and/or in sequence based on the modules' inter-dependencies. The sample bot uses a CoBot named entity recognition API along with open source noun phrases and co-reference models in the NLP pipeline to extract features from user utterances. After the NLP pipeline completes, the NLP features are used to select a set of response generation modules that run in parallel to produce candidate responses that flow into the response ranker for final selection of the best response. The response generators in the sample bot made use of the AlexaTM, knowledge retrieval, news, and neural generator APIs, as well as our API for open-domain QA that accesses Alexa's semantic knowledge graph (also used in production to answer Alexa questions). Ranking was done using the ranker model to augment a simple priority-based ranking of responses. Additionally when a device supported APL, multimodal templates and commands were generated along with the spoken responses. When APL was enabled we also provided examples of special handling for multi-turn touch-based and voice-based responses in the ranking and selecting strategies of the dialog manager.

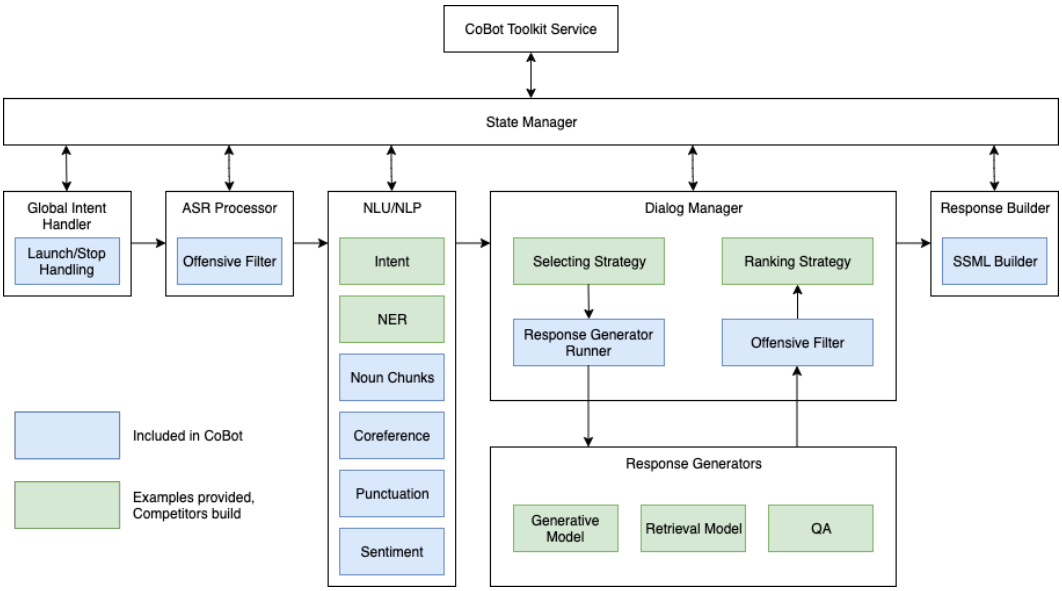

Figure 6: CoBot Architecture

We wanted this year's sample bot to remain as simple as possible while still illustrating all of CoBot's capabilities. Past example SocialBots provided in the toolkit had been more complicated, causing teams to spend significant effort trying to understand the provided example before ultimately building their own bot from scratch. This year's sample bot demonstrates how teams can use CoBot's APIs to build a working bot, while exhibiting readable, high quality code that is straightforward to understand.

2.2 Automatic Speech Recognition and Text to Speech

We provided Automatic Speech Recognition (ASR) to convert user utterances to text and Text-To-Speech (TTS) to render text responses from SocialBots to users via voice. Our ASR model is tuned for conversational data and features custom end-pointing and extended recognition timeouts for longer free-form interactions. Alexa Prize teams also received access to tokenized n-best ASR hypotheses, including confidence scores for each token, as well as voice-based Sentiment scores (activation, valence, satisfaction) generated in real-time. For TTS, all teams are required to use the standard

Alexa voice; however they have the ability to use Speech Synthesis Markup Language (SSML)[6] to control how Alexa generates the speech. For example, SSML can be used to add custom pauses within the Alexa response or add an "excited" emotion to Alexa's voice.

2.3 Infrastructure

We provided free Amazon Web Services (AWS) to teams, including but not limited to: GPU-based virtual machines for building models, SQL/NoSQL databases, and object-based storage with Amazon S3. We also provided load testing and scalability tools and architectural guidance.

2.4 User Feedback Data and Evaluation Metrics

A key benefit provided to the SocialBot teams was the ability to field their bots with Alexa users. After interacting with a SocialBot, users were prompted for satisfaction ratings and feedback on their experience. Each SocialBot team had access to these metrics and also received an anonymized leaderboard daily that presented average metrics and rankings for all participating bots. In addition, teams were provided with transcriptions of the free-form feedback shared by users at the end of their interactions with the team's bot, allowing the teams to gain qualitative insights into the users' impressions of the SocialBots.

2.5 Support from the Alexa Prize team

In addition to providing data, infrastructure, AI tools and models, we engaged with university teams in several ways to provide support and feedback:

- A virtual pre-Bootcamp to on-board university teams to the CoBot Toolkit and prepare teams for Bootcamp.
- A hands-on Bootcamp with training materials and best practices design guidelines.
- Virtual sessions with university teams on CX design, model training and evaluation, and competition guidelines to prepare teams for each phase of the competition.
- An internal beta phase, to provide traffic and feedback from Amazon employees to help inform and improve bot performance before general availability to all Alexa users.
- Detailed report on bot experiences prior to public launch, evaluating functionality as well as the bot's ability to maintain anonymity and handle inappropriate interactions.
- Bi-weekly office hours over the course of the competition for consultations with a dedicated Program Manager and members of Alexa science and engineering teams. The university team members discussed issues and observations over an aggregate 170 hours of demo sessions.
- On-demand access to Alexa Prize personnel via Slack and email.

3 Scientific Advancements

3.1 From the Alexa Prize participants

Over the course of the challenge, participants made numerous improvements to their bots in order to improve their performance and enhance the overall user experience. Scientific innovations and engineering optimizations spanned multiple areas including (1) adoption of large language models (LLMs), (2) the development of a dialog control layer to prompt and control the LLMs, (3) use of an LLM as a 'jack of all trades' for many different tasks, (4) synthetic data and corpus generation using LLMs, (5) extensions to knowledge-based approaches, (6) automated dialog evaluation, (7) optimizations to mitigate LLM latency, (8) innovations addressing responsible AI, and (9) a broad spectrum of different approaches to providing a compelling multimodal user experience for open-domain dialog. Teams reported how the scientific innovations impacted user experience, measured using Customer Satisfaction Ratings (CSAT) score, as well as evaluating performance of specific modules on external public datasets such as Daily Dialog [32] and Topical Chat [15].

[6] `https://developer.amazon.com/en-US/docs/alexa/custom-skills/speech-synthesis-markup-language-ssml-reference.html`

3.1.1 Large Language Models

Large language models (LLMs) are far from new to the SocialBot grand challenge. They played an important role in extending the capabilities of bots as neural response generators over the last two instances of the challenge [21, 14]. GPT-2 [41] based neural generators trained on Alexa Prize data have been made available to teams for several iterations of the competition. This is nothing however compared to the level of use of LLMs in SGC5. All teams actively integrated LLMs into their SocialBots with popular models including Vicuna (7B/13B) [5], BlenderBot 3 [47], FLAN-T5 [6], Falcon 40B [2], and Alexa Teacher Model (5B/20B) [49]. In addition to increased capacity the massive uptick in adoptions of LLMs has been driven by the availability of models that support instruction prompting and in-context learning. With rapid changes happening in the field, we saw teams move quickly to experiment with and adopt new models as they were released across the course of the competition.

In early iterations of the challenge, in order to achieve competitive performance in user ratings teams relied on authoring hand crafted and carefully curated response flows for common topics, and generally adopted neural generation as a fallback for unsupported topics or scenarios. SGC5 is the first time we have seen highly performant SocialBots with little to no hand curation. Team GauchoChat (UC Santa Barbara) [51], for example, used Vicuna 13B as the primary response generator. Team HokieBot (Virginia Tech) [45] and Team NAM (Stevens Institute of Technology) [33] also did not invest in building hand crafted responders, instead focusing on LLM performance. Limitations remain however with respect to support for discussion of current events, personality, and providing opinions.

It is important to note though that these models are not used 'raw', that is taken as is and simply prompted with the raw dialog context. They are fine-tuned to improve their performance on social conversation and/or controlled using carefully crafted instruction prompts. For example, Team Alquist (Czech Technical University) [25] identify several shortcomings with BlenderBot 3, such as small fraction of conversational data in the training set, and high latency. They develop Barista, in which they retrain some of these modules with more balanced datasets and using smaller models, and achieve higher accuracy and lower latency. They also fine-tune Vicuna 7B and include it as one of the LLMs in their overall system. Similarly, Team NAM (Stevens Institute of Technology) [33] fine-tuned BlenderBot and BlenderBot 2 [47] models on a custom dataset created by prompting a large language model, resulting in an increase in CSAT of $\sim$0.4-0.5.

3.1.2 Dialog control layer around LLM

One repeating pattern we have seen in the competition is teams making use of a large LLM as the primary responder, but wrapping that LLM with a dialog control layer which dynamically adjusts the prompt used with the LLM. One of the motivations for this, pointed out by Team GauchoChat, is that contemporary LLMs are trained to be *reactive* rather than *proactive*. They are trained to take instructions and respond to the user's input, but they are not trained to take initiative and draw out the user, suggesting topics and taking the control of the conversation when the user is losing interest or lost for words. These properties are critical for the creation of effective and engaging open-domain social dialog systems.

One example of this 'dialog layer on LLM' approach can be seen in Team GauchoChat's (UC Santa Barbara) [51] approach to prompting Vicuna 13B to drive social conversation. They argue that no single prompt can be expected to trigger the best responses for all users and personalities. They author a set of 20 carefully selected prompts that can be thought of as moves to take in the open domain dialog (e.g. "Share a thought provoking quote and ask the user for their interpretation", "Offer an empathetic and supportive response to make the user feel valued", or "Share a personal fact"). They then train an 'LLM Promptist' starting from a pre-trained RoBERTa [34] model and use reinforcement learning with user ratings as the reward in order to learn a policy to select prompts during open domain dialog. With the introduction of this feature in their SocialBot in the Initial Feedback Period they saw a L3d CSAT improvement of approximately 0.5. Another related innovation is their approach to dynamic topic switching. Team GauchoChat developed an approach which identifies when a topic has been active for some time and the user is showing low engagement. In these cases they retrieve content for a new topic and dynamically change the topic by manipulating the prompt used with Vicuna 13B. In addition to using Vicuna for the generation step they also use it for classifying user disengagement and for formulation of a web search query to retrieve content to discuss. They found a correlation between successful topic switches driven by the dynamic mechanism and higher CSAT

ratings and observed an increase in L3d CSAT of ∼0.3-0.4 after the introduction of the feature in the Semifinals Period. They also report a correlation between dialogs with successful topic switches driven by this mechanism and 4.0-5.0 ratings.

Another example of this pattern of a dialog layer and formulation of prompts to feed to an LLM is Team HokieBot's (Virginia Tech) [45] PersonaDial. They argue for explicit modelling and tracking of user preferences (e.g. likes and dislikes with respect to movies and artists) outside of just keeping the raw dialog context. The key motivation is given limitations on the length of context that can be fed into the LLM, it is difficult to track all topics covered across a long conversation and the user sentiment towards them. They use an LLM to create and distill a synthetic corpus of 7k conversations spanning 44 topics annotated with topic chains and user preferences. This is used to train a series of models that enable tracking of user topics and formulation of prompts. A Memory Generation module (using InstructDial [18]) is used to extract topics from the latest turn in the conversation and user preferences towards those topics. A Guidance Generation module (fine-tuned FLAN-T5 large [6]) is used to formulate a prompt based on the memory. This is then fed to the final module Response Generation (BlenderBot 400M distilled) in order to formulate a response to provide to the user. These three models operate in concert to provide more engaging responses tailored to the personality of the user. The team provide qualitative examples of improvements in responses compared to direct use of BlenderBot, but do not present results showing impact on CSAT.

Team CharmBana (University of Illinois) [43] also address the need for SocialBots to be more *proactive*. They address the issue of users being passive with an approach that uses social commonsense reasoning to identify related topics to the current conversation and then uses these to formulate a search query to retrieve content that can be fed into an LLM prompt for response generation. They first identify the current main topic using an instruction prompt to a FLAN-T5 [6] model. They then prompt a social commonsense model COSMO [24] trained on SODA, a corpus of 1.5M synthetically generated social dialogs. This step will for example identify the fact that if we are talking about a movie it makes sense to extend the conversation with discussion of reviews of the movie. They then use an instruction tuned FLAN-T5 model to take the dialog context as input and generate an appropriate search query to retrieve content e.g. movie reviews. Retrieved content is then fed along with dialog context to the response generation component to generate a compelling prompt that incorporates content retrieved from search. Evaluation with human judges shows improvements in response quality on various dimensions compared to direct use of BlenderBot 3 [47] or FLAN-T5 without the use of the COSMO model for social commonsense reasoning.

Team Alquist's (Czech Technical University) [25] Barista framework combines both a fine-tuned version of BlenderBot 3 and a fine-tuned version of Vicuna 7B [5] (Vicuchat). A key aspect of their approach is the use of a classifier to determine whether search is required, which runs in parallel with prompt-based search query generation. Search is only conducted if required in order to reduce latency and if used, retrieved content is fed along with the dialog context to the BlenderBot 3 model to generate a response. Another feature of their approach is hybrid dialog, where based on a dialog tree a topic or fun fact is introduced to the conversation, and then the LLM responder is allowed to continue the conversation from there for several turns.

3.1.3 LLMs as the workhorse for many tasks

Another theme that was evident in the SGC5 competition was that teams used LLMs not just for neural response generation based on dialog context, but as a general purpose workhorse for just about any machine learning task needed to support their bots. Some teams focused on a single model and used it over and over with carefully engineered prompts for different tasks. For example, Team GauchoChat (UC Santa Barbara) [51] used Vicuna 13B to detect signs of boredom from the user, to identify new potential conversation topics, to formulate search queries for retrieving external knowledge, to generate customized prompts for response generation based on the user and context, to generate image search queries based on dialog context, as well as to perform the main response generation. Team HokieBot (Virginia Tech) [45], in contrast, use three different models (InstructDial, FLAN-T5, and BlenderBot) to power the three different aspects of their PersonaDial capability. Similarly, Team CharmBana (University of Illinois) [43] combined FLAN-T5, COSMO, and BlenderBot in their approach to generation of engaging responses to draw in passive users. As discussed in the following sections, in addition to these online uses many teams used LLMs offline for the creation of synthetic corpora and knowledge.

3.1.4 Synthetic data and corpus creation

Another recurring theme was the use of large language models offline to create synthetic datasets. Team NAM (Stevens Institute of Technology) [33] prompted various large language models for different styles, topics, scenarios, intents, and personalities to create a diverse dataset, which was then used to fine-tune BlenderBot and BlenderBot 2 response generation models. Team HokieBot (Virginia Tech) [45] generated 4,000 synthetic conversations across 44 different topics using a large language model, which was used to fine-tune a topic classification model. Team Alquist (Czech Technical University) [25] prompted a large language model to act as a badly behaved user, providing data which was then used to analyze safety classifiers and improve responsible AI aspects of their bot. Other teams used various large language models to generate silver labels for datasets for training classification models. For instance, Team CharmBana (University of Illinois) [43] used labels generated by a large language model for topic classification to fine-tune the Flan-T5-large model, and Team Tartan (Carnegie Mellon University) [31] generated scores for conversations using a large language model, which were subsequently used to train a response ranking component. Team Athena (UC Santa Cruz) [13] prompted a large language model to generate sets of personal questions and potential follow-ups based on user models. They use this to train a personal question generator which on the basis of a 15 day A/B test they found to improve both user ratings and dialog length.

3.1.5 Knowledge-based approaches

While the focus in this iteration of the competition was on prompting LLMs to generate responses, several teams used knowledge graphs or dialog flows or trees to drive parts of the conversation in concert with LLMs. Team Alquist (Czech Technical University) [25] make use of a library of dialog trees supporting system initiative dialog on a broad range of topics. These trees are integrated with LLMs in a number of ways. First of all, out of domain utterances, identified by an intent classifier, are passed to a loop that allows the LLM to respond for several turns. Secondly when the user takes initiative with proactive questions the LLM is engaged. Thirdly, a hybrid dialog mechanism is employed where a carefully curated engaging question, fun fact, or comment is used to start a conversation segment and then control is passed to the LLM. Lastly, locations in dialog trees with short acknowledgements and less informative responses are used as entry points to LLM-based interaction.

Team Athena (UC Santa Cruz) [13] employed an approach called FLOW-RG which enables interleaving of mini-flows to provide continued interaction on a specific topic [40]. The mechanism was expanded with additional universal templated mini-flows that support declarative specification of content for personal questions, informal trivia, and personal opinions. Team Athena also employ knowledge graph driven responders to identify ways to extend conversation on a topic by navigating to related entities and relations in the Wikidata graph [40]. They expanded on this approach by moving from template-based to neural generation, using prompt-based transfer learning and self training to extend dialog act coverage from conversations on video games to movies, music, TV, and sports, and setting up cross domain dialog policy to drive knowledge graph based dialog.

Team Chirpy Cardinal (Stanford) [4] introduced a new declarative domain specific language (CAMEL) for authoring dialog graphs in order to improve interpretability, maintenance, and flexibility. In order to further accelerate dialog graph development and combine the benefits of declarative approaches with LLMs, they also introduce a hybrid approach (Goldfinch) which automatically grows the coverage of their dialog graph by prompting a large capacity LLM. Specifically they introduce a series of techniques including prompting a large language model to widen and deepen the knowledge graph. This approach reduces the burden of dialog authoring by leveraging knowledge embedded in the LLM, but in contrast to directly using the LLM to generate responses at runtime, it produces a human inspectable resource that can be edited and corrected. In addition to enabling strong responsible AI checks on system output, this approach also significantly reduces latency. They generated a graph with over 2000 nodes to drive open-domain conversation and report 3.59 CSAT for conversations that reach the most general automatically generated nodes in their graph vs. 3.24 for those that do not.

3.1.6 Automated Dialog Evaluation

Another area in which teams innovated was in the creation of automated methods for evaluating both individual turns and whole dialogs. Team HokieBot (Virginia Tech) [45] argued for the need

to evaluate dialog over multiple different fine-grained dimensions, such as fluency, coherence, politeness, and engagingness. They propose InstructEval, an instruction tuned model for predicting these dimensions based on prompts. They start with the Flan-T5-large LLM [6] and instruction tune it over a collection of 69 diverse tasks including various kinds of evaluation (scoring, ranking, comparison) alongside summarization and dialog tasks. They evaluate InstructEval by examining the correlation of its outputs to human judgements on the Topical Chat dataset [15]. They meet or exceed the performance of a range of other dialog evaluation metrics including SOTA performance from the G-eval [35] approach using prompting of large capacity language models to conduct evaluation. They also evaluate on the FED dataset [36] and show the approach outperforms other approaches on prediction of previously unseen evaluation dimensions. They then incorporated this measure into their approach to response ranking.

Team Tartan (Carnegie Mellon University) [31] looked at how user expectations correlate with the duration of the conversations. Specifically, they show that users that invoke the system with the SocialBot invocation as the launch phrase usually have much lower early exit rate than those that invoke the system with more generic phrases including words like "chat" and "conversation".

Team CharmBana (University of Illinois) [43] also innovated in dialog evaluation. They developed an extension of the FED approach to unsupervised reference free evaluation [36]. They replace the FED scoring function with conditional point-wise mutual information (C-PMI) function in order to capture interaction between the response and context. This resulted in a 60% relative improvement in correlation to human judgements compared to FED, with particularly strong improvements in key dimensions for social dialog such as Interestingness and Engagingness.

3.1.7 Mitigating LLM latency

LLMs provide tremendous capabilities for responding robustly to a broader range of inputs from users in open dialog and performing numerous tasks needed to build SocialBots. They do, however, pose significant challenges for maintaining acceptable levels of latency for a spoken dialog system where it is necessary to operate within a quite tight latency budget in order to provide a compelling experience to users. This will continue to be an issue even as more powerful inference architectures become available, as in parallel the size and capabilities of models is likely to increase. Several teams, including Team GauchoChat (UC Santa Barbara) [51] and Team CharmBana (University of Illinois) [43] developed techniques to reduce perceived latency using progressive responses. In these approaches, a quick initial response is provided, and while this is being spoken a larger LLM has an additional several seconds to generate tokens. Team Chirpy Cardinal (Stanford) [4] realized that the largest window of time available to a SocialBot is when the bot is talking and then the user is thinking and responding. They generate a list of likely responses to the previous prompt and feed these in a batch to a large capacity LLM (Falcon 40B [2]) in order to pre-generate candidate responses. If the actual user response is sufficiently similar to one of the predicted potential responses, the bot is able to respond immediately with a higher quality answer. Otherwise it backs off to a response generated by a smaller, faster model (BlenderBot 3).

3.1.8 Response Ranking and Selection

A common pattern across SocialBot architectures is to apply a range of different competing responders on each user input. A key technical challenge then is to select among these competing responses. In SGC5 we provided the teams with access to an API providing a default BERT based response ranker trained on previous Alexa Prize data [20]. Teams also innovated and extended on the response ranking capability. Team Tartan (Carnegie Mellon University) [31] created a custom dataset using Daily Dialog [32] and prompted a large language model to generate target scores. They train a RoBERTa-based model on this dataset and apply it to rank responses. Team NAM (Stevens Institute of Technology) [33] use an ensemble of the Alexa Prize default BERT ranker and scores from DialoGPT [55]. For response ranking, Team HokieBot (Virgina Tech) [45] used a combination of scores including their InstructEval instruction tuned turn-level evaluator, UniEval [56] multi-dimensional evaluation, and the provided BERT based ranker.

3.1.9 Responsible AI

Maintaining the principles of responsible AI is a significant concern for open-domain dialog systems where users may frequently bring up controversial topics and there is risk that the system engages

on the topic and provides unsafe responses. Several teams showed innovations specifically directed towards this challenge.

Team Thaurus (University of Madrid) [11] make the point that strategies which redirect users to alternative topics (e.g. "I'm not comfortable talking about that.") when they bring up a controversial topic or use swear words may result in the user feeling that they are not heard and lead to disengagement. The take inspiration from Eliza [53] and allow expression of user opinions while avoiding providing personal opinions. Their approach involves developing a more fine-grained model for identifying types of toxicity and the target of the comment. They distinguish neutral vs. controversial vs. sexual comments and opinions and whether these are directed towards the SocialBot or not. The outputs of the model are used to drive a custom responder for handling toxic inputs. They found that in cases where the module would be triggered the mean user rating was 3.328 with the new module vs 2.949 without.

Team Alquist (Czech Technical University) [25] also address responsible AI in their work on dialog safety. They first trained classifiers on datasets providing examples of hate speech, bias, and offensive language. They then used a large language model to simulate toxic conversations from a user and used those outputs to drive conversation with several large dialog-oriented language models including DialoGPT [55] and BlenderBot [47]. They found that a considerable proportion of the responses produced (10-20% overall) were identified as potentially containing unsafe content. They also found that about one third of the flagged responses were false positives. To address this, for their SocialBot they developed a combined safety filter which combines automatic classification with a second rule-based module. In evaluation they showed significant improvements in F1 score with the combined approach.

3.1.10 Multimodal Interaction for Open-Domain Dialog

While at their origin conversational assistants focused on a voice-first user experience, the context of interaction they appear on (e.g. smart speakers with displays, mobile phones, smart TV, automotive) is increasingly multimodal. Recognizing this shift, SGC5 is the first SocialBot challenge to require teams to provide a multimodal user experience for their competing bots. Teams were required to develop experiences that are compelling both on voice enabled devices such as Echo Dot and multimodal devices such as Echo Show or FireTV. For the purposes of the finals event, judges evaluated interaction with SocialBots running on Echo Show 8 devices.

There is a long history of multimodal interfaces for task-oriented dialog [50, 10] for tasks such as local search or travel booking, but there has been considerably less attention to multimedia elements for open-domain dialog where the user engages with the bot in conversation on any topic. The availability of a screen drove teams to develop numerous innovations, as shown in Figure 7. Teams leveraged the display to provide a more compelling presentation of the system's textual response. Team Alquist (Czech Technical University) [25] developed a 'karaoke' like display where the UI presents highlighted and/or scrolling text as synchronized with the voice prompt. This solved a major issue of how to display longer SocialBot responses. The scrolling effect was then integrated into the CoBot Toolkit and provided to all of the teams as a standard asset.

Teams Alquist (Czech Technical University) [25], Athena (UC Santa Cruz) [13], Chirpy Cardinal (Stanford) [4], GauchoChat (UC Santa Barbara) [51], and Thaurus (Technical University of Madrid) [11] also used the display to allow for visual presentation of multiple choice options (for example, for selecting a topic to discuss) and for prompting the user with potential continuations of the dialog.

Teams also augmented the conversation with images and video content. Given recent advances in quality and availability of generative AI for image generation, several teams utilized prompting of multimodal language models in order to generate images either online or offline. Team Thaurus (Technical University of Madrid) [11] used a text-to-image model to generate multiple images from extracted entities and then chose the best image using two approaches: 1) Comparing the text embedding used to generate the image and a caption generated from the image and 2) using a multimodal encoder to directly compare the embeddings from the prompt and the generated images. Team Thaurus found the second method to be most effective. Team GauchoChat (UC Santa Barbara) [51] and Team Athena (UC Santa Cruz) [13] extracted entities from recent turns and used these to render an image using a text-to-image model. If the generation time exceeded the latency budget, the image was cached for future use.

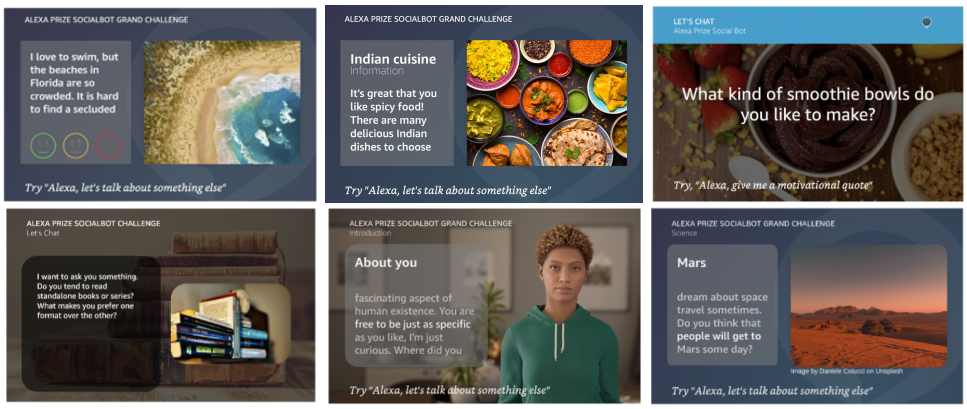

Figure 7: SocialBot on Multimodal Devices

One of the risks of online generation of images is the quality or appropriateness of the rendered image. Some teams, including Team Chirpy Cardinal (Stanford) [4] and Team Athena (UC Santa Cruz) [13] pre-generate images for popular topics from conversation logs and manually verify them for correctness. Chirpy Cardinal adds 'watercolor' to the prompt for a more relatable feel and Team Athena uses '4k photo realistic' to generate the appropriate size and style.

Team CharmBana (University of Illinois) [43], Team Alquist (Czech Technical University) [25], and Team GauchoChat (UC Santa Barbara) [51] all explored using live resources to retrieve images at interaction time (Pixabay and Bing). Relevance and appropriateness of images is guided by including directions such as 'child friendly' in the search query.

Several teams experimented with the addition of various types of avatars to enhance the conversation. Team Alquist generated 3D rendered Avatar videos that can be played in the background to make the conversation more engaging. It was not technically feasible to integrate real-time visual TTS, but they augmented the conversation with videos for particular response categories such as neutral/idle, greetings, and dialog error. The team reported an improvement in CSAT from 3.25 to 3.40 on multimodal devices after introduction of their avatar along with significant increases in average conversation duration.

3.2 From the Alexa Prize Team

This section provides details on the models behind the new APIs provided by the Alexa Prize team to the university teams for SGC5.

3.2.1 Response Ranking

For SGC5 we added to the components provided to teams a trained response ranker for selecting among responses from different responders. To build this we first constructed a new dataset RSD (Response Selection Dataset) for response selection by showing human annotators multiple response candidates produced by different response generators for a given turn and dialog context, and asking them to annotate all responses that are appropriate for that specific dialog context.

We directly fine-tune a BERT-base [8] on the RSD and refer to this model as BERT-Ranker [20]. To construct our input, we concatenate the dialog context with a system response and follow the same training procedure used, which uses the pooled output representation by the BERT model, passes it through a linear layer followed by a sigmoid function, and minimizes the binary cross-entropy function to predict whether the given system response is positive or negative.

We find that our model trained on RSD significantly outperforms those trained on existing datasets showing the benefit of bringing in human annotated data for this task. Additionally we demonstrate that both strategies of using multiple positive candidates and using manually verified hard negative candidates can bring in significant performance improvement in comparison to using adversarial training data, e.g., an increase of 3% and 13% in Recall@1 score, respectively.

3.2.2 Contradiction Detection

Contradictory responses across the course of a conversation are a significant problem for LLM-based systems and more generally for open-domain dialog systems that combine multiple responders. Contradictions can confuse the user and erode trust. To help address this we developed a contradiction detection capability and made it available to teams through CoBot APIs.

We formalize dialog contradiction detection as an NLI task. Given a list of utterances $x = \{u_1^H, u_1^B, ..., u_n^H, u_n^B\}$ representing a dialog, the task is to determine if the last bot utterance u_n^B contradicts any previously conveyed information contained in the past bot utterances $\{u_1^B, ..., u_{n-1}^B\}$. Note that we are using human and bot alternating turns here (referred to as H and B), but they can be human-human conversations too. We pair every past bot utterance with the last one, and then feed each pair to the classification model, i.e. RoBERTa-Large [34], for both training and inference. For training data, we combine the DECODE dataset [38] and our in-house annotated data based on DSTC9 Track 9 interactive dialogs [37]. In total, we have 13,965 and 15,084 positive and negative samples, respectively. In terms of performance, we have evaluated the developed contradiction detection model on the benchmark DECODE test set, which results in 84.89% in accuracy, 92.83% in AUPR, and 89.85%/78.67%/83.89% in P/R/F1 scores (decision probability threshold is 0.1).

3.2.3 Utterance Rewriting

Utterance rewriting rewrites a user or system utterance based on context in order to resolve co-reference and ellipsis and make it more explicit. The utterance rewriting model is a seq2seq model (BART [30]) that can rewrite a user or a system utterance based on context to resolve co-reference and ellipsis. The input is the concatenated context utterances and the original last utterance, with special tokens inserted before each utterance to indicate its speaker. And the output is the rewritten last utterance. It has been fine-tuned on our collected data which consists of 40K training and 3,200 test samples on open-domain conversations [22]. In terms of performance, the fine-tuned BART-Base model with greedy search decoding can achieve 0.657, 0.7521, 0.8527, and 0.2072 in terms of BLEU4, ROUGE-2, ROUGE-L, and EM scores.

3.2.4 LLMs for Neural Generation

For the first time in SGC5, we provided teams with higher capacity LLMs based on the Alexa Teacher Model [49] (ATM) a sequence-to-sequence model conversationally pre-trained to improve performance on dialog and response generation tasks. In addition to ATM models, we also provided updated and controllable versions of the smaller GPT-2-based [41] models provided to teams in earlier iterations of the challenge.

3.2.4.1 Alexa Teacher Model

We provided three versions of the Alexa Teacher Model [49]. In order to facilitate easy experiments with in-context learning methods or the use of these LLMs withing a complex system, we provided these models as APIs as described in section 2.1.4 The AlexaTM 5B and AlexaTM 20B models were conversationally pre-trained to improve response generation.

Conversational pre-training of the vanilla AlexaTM models required the following two steps:

- The first step was pre-training on a conversational dataset that was filtered to exclude profanity and non-English data, which after filtering contained 1.5TB of data.
- In the second step, the model was fine tuned on the following datasets: Wizard of Wikipedia[9], Wizard of Internet[26], Blended Skills Talk[48], Empathetic Dialogues[42], Commonsense Dialogue[58], Persuasion for Good[52], Topical Chat[15], Edina Corpus[12][27], and WikiDialog (100k random subset)[7].

The ATM 5B model is hosted on g5.4xlarge EC2 instances and the 20B models are hosted on g5.12xlarge EC2 instances. All models are converted to bfloat16 for running inference. The 5B models use a single GPU per model instance and the 20B models use 2 GPUs per model instance. The models were hosted as CoBot remote Docker modules, the details about which are provided in Section 2.1.2 Large Language Model Support in CoBot. Finally, API gateways were used to manage traffic to these services.

3.2.4.2 Updated GPT-2 neural generators

We updated our GPT-2-based neural response generators (NRGs) by training on more recent SocialBot dialogs from the Alexa Prize bots. We first applied a set of dialog-level filters on all SocialBot logs from January to December 2021 including user ratings, dialog length, and duration, and only trained our NRG models using filtered data. The models were trained to take in the dialog history and minimize the cross-entropy loss based on the ground-truth response. We also reserved a small set of SocialBot data with turn-level error annotations as well as the newer logs from January to May 2022 as two test sets to evaluate response quality. To mitigate the common issue of abrupt topic switches in multi-turn conversations, we have also investigated several topic filtering methodologies to remove incoherent and problematic turns that can hurt overall dialog quality. In addition to the updated NRG model we provided two more variants: Topic NRG and Empathy NRG, described below, which enable control of response generation based on a dialog policy.

It was shown in [19] that being able to control the content or style of a response generated from a neural model can enable more appropriate responses. We can craft a dialog policy that can help best leverage this controllability during the course of the conversation. We crafted a dialog policy to address off-topic responses. We first trained a neural generation model where we can control the topic of the generated response. Our SocialBot logs had topic annotations from humans. This included coarse grained topic labels such as Politics, Sports, Music, Movies and a catch-all Other category. Similar to the conversational pre-training described above we trained a GPT2-XL model on our SocialBot logs, but in addition to sending in the dialog context we also sent in a special token representing one of our coarse grained topics. This special token refers to the topic of the ground truth response. This token was a separate embedding that was randomly initialized and learned during training. Our hand-crafted dialog policy involves predicting the topic of the last user utterance and trying to generate a response that is on the same topic by including the special token that corresponds to that topic. The resulting API allows a dialog manager to control the topic of the generation by providing an appropriate topic token.

Empathy NRG also aims to control style of generated responses, but specifically tries to generate an empathetic response when appropriate. To build this, we adopted an off-the shelf empathy classifier [57] and ran it over our SocialBot logs to assign empathy labels. For each response from a SocialBot we predict if this response is empathetic or not. We then follow the same training procedure described above but this time the special token refers to empathy. A dialog policy can then explicitly request empathetic responses from the NRG.

3.2.5 Automated Evaluation: Open-Domain Dialog Evaluation Signals

We released the Open-Domain Dialog Evaluation Signals (ODES) user utterance classifier [29] to the teams to provide some additional automated dialog evaluation metrics for their conversations. The ODES model is a RoBERTa [34] based classifier for automated dialog evaluation. It is trained on a dataset curated by filtering and annotating Alexa Prize conversations. We defined a set of utterance level categories that capture indicators of dissatisfaction where the user expresses that the system did not understand them, they do not understand the system, they are disinterested in the content, or when they explicitly call out the system for contradiction, repetition, or changing the subject. The categories also include obscene utterances, insults, and compliments along with requests to stop the interaction and requests to change topic.

The ODES classifier is part of a broader suite of evaluation metrics that are combined to predict Customer Satisfaction Rating (CSAT), as described in [29]. In the approach outlined in that work, there is a baseline model that takes ODES utterance level results, acoustic sentiment scores, ASR confidence, and a number of language model derived features including FED metrics [36] and DialoGPT-based measures of relevance and specificity [55] and combines them at dialog level using an multi-layer linear perceptron regressor to predict user or annotator ratings. We further extended that work, adding a Counterfactual-LSTM (CF-LSTM) that models the relation of turn-level features to dialog-level using an LSTM and applies different regression models to calculate the final predicted rating depending on which ODES feature categories are found in the dialog. This approach produced significant improvements in correlation with user assigned ratings compared to the baseline model.

4 SocialBot Performance: Discussion

The primary mechanism for evaluating the progress of SocialBot teams is the Customer Satisfaction Ratings (CSAT). After each interaction, Alexa users were asked to rate their interaction with the SocialBot on a scale of 1-5. This rating was aggregated for each team on a daily basis (L1D) and a weekly basis (L7D). The scores were used by the Alexa Prize team to advance university teams from beta period and into the semifinals and finals periods. University teams used this score to compare individual conversation quality and improve their SocialBot architectures and retrain their models.

It's important to note that the SocialBot rating prompt differed from the prompts used in the SimBot and TaskBot competitions, and thus, the ratings should not be directly compared between the different competitions.

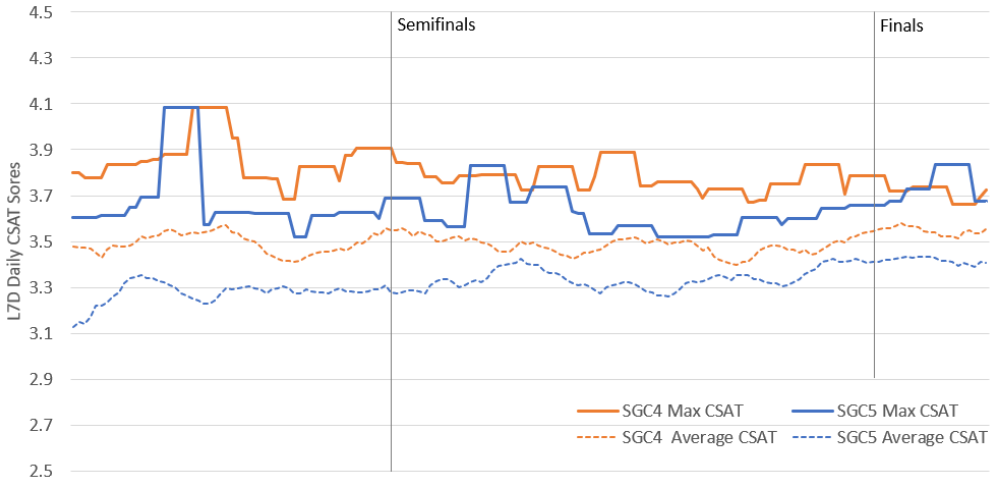

Figure 8: L7D Average CSAT Rating and L7D Maximum CSAT rating over the period of the competition for SocialBot Grand Challenge 4 and SocialBot Grand Challenge 5 finalist teams.

Since the competitions were of slightly different duration (SocialBot Grand Challenge 4 semifinals began on 5/4/2021 with the finals event beginning 7/26/2021; SocialBot Grand Challenge 5 semifinals began on 5/8/2023 with the finals event beginning on 8/1/2023) for the purpose of Figures 8 and 10 the timelines are aligned on the start of the finals event. These two charts also include the data from the finalist teams only.

The average CSAT score for teams differed by only 0.1 to 0.2 L7D CSAT for most of the competition (Figure 8). The maximum CSAT was calculated by taking the top L1D CSAT score for any of the finalist teams over a 7 day period. The resulting comparison compares the best user experience across both competitions.

When reviewing the L7D Maximum CSAT (Figure 8), the best user experience was at parity early in the competition. In the weeks leading up to semifinals the teams focused on the stability of their SocialBots in order to meet CSAT advancement criteria. The maximum CSAT plateaued for teams this year until just after the beginning of the semifinals period where there was a jump in the quality of the conversations.

Immediately after the start of semifinals teams had begun to implement stronger conversations (Section3.1.5 Knowledge-based approaches) and were experimenting with the integration of research LLMs (Section3.1.1 Large Language Models) and integrating multimedia (Section3.1.10 Multimodal Interaction for Open-Domain Dialog). Since the finals event was designed to feature the multimodal experience, it was imperative the finalist teams improve the interactions on a screened device. This required a deeper focus on the underlying SocialBot architectures (Section3.1.2 Dialog control layer around LLM) and shifted the focus away from chasing user satisfaction. As the Semifinals approached the final 14 days, the teams returned their focus to user satisfaction in order to gain a place in the finals event. As the teams passed the finals event, the quality of the user experience surpassed the L7D Maximum CSAT scores from the 4th SocialBot Grand Challenge.

4.1 New SocialBot Teams

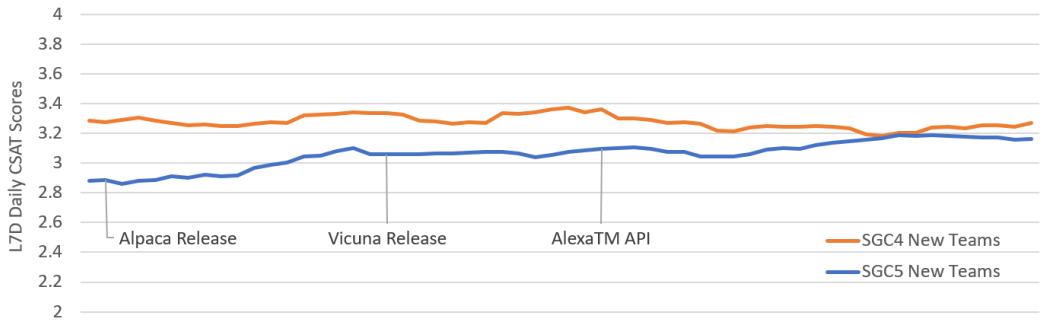

Figure 9: L7D Average CSAT Ratings for New SocialBot Grand Challenge 4 and New SocialBot Grand Challenge 5 teams from the initial feedback period until the start of semifinals.

While teams that have competed in previous competitions have infrastructure from previous years, new teams start with only the tools provided in the CoBot toolkit. Unlike the previous competition, new teams this year started this competition attempting to integrate publicly released research LLMs (Section3.1.1 Large Language Models) into their dialog management strategies. The initial CSAT ratings gap (at the start of the initial feedback period) compared to the previous year (Figure 9) was 0.3 points due to this engineering effort. This year's new teams gained ground and reached parity with the previous competition's new teams before semifinals.

4.2 Conversation Duration

Our primary success metric for conversation duration is the p90 duration, or the 90th percentile duration of conversations (i.e. 10% of conversations have a duration longer than this number). We consider this a strong proxy for a maximum duration that the SocialBot can sustain an interesting and engaging conversation with a dedicated interactor.

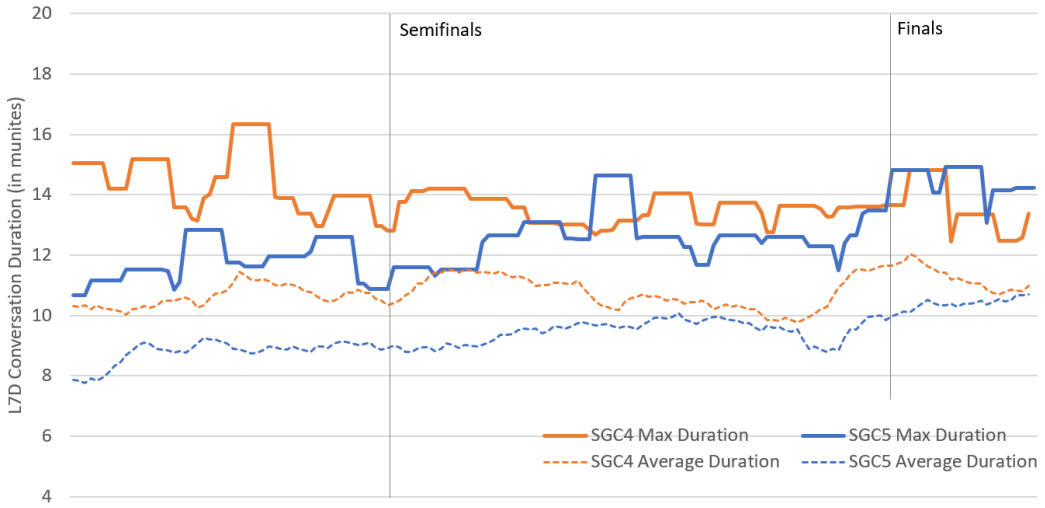

Figure 10: Conversation duration for the top 10% of conversations for the SocialBot Grand Challenge 4 and SocialBot Grand Challenge 5 finalist teams.

Throughout the 5th SocialBot Grand Challenge, the L7D Maximum Duration varied when compared to the prior year's competition. The maximum duration was lower before semifinals this year but surpassed the previous year's scores mid-semifinals. (Figure 10). By the finals event this year's maximum duration met and exceeded that of the previous competition.

4.3 Multimodal Interaction

As previously discussed, the SocialBot Grand Challenge has traditionally been focused on the voice experience. The 5th SocialBot Grand Challenge incorporated multimodal conversations (Section2.1.3 Multimodal Experience), with the finals event being held exclusively on multimodal devices. At the start of the finals event, the top 7-day average rating achieved by a SocialBot was 3.50 overall with a conversation duration in the top 90th percentile of 9 minutes 8 seconds. To highlight the importance of integrating multimedia into the conversations this year, for that day the top L7D average rating by a SocialBot for multimodal conversations was 3.54 with a 90th percentile conversation duration of 14 minutes 27 seconds.

During the Semifinals period we introduced homecards, which provide an advertisement promoting the challenge to multimodal users on screened devices. This boosted multimodal traffic to almost 50% of the total traffic. As we neared the finals event, the amount of homecard advertising was reduced to allow traffic to be directed towards other concurrent competitions. This resulted in the final slow decline in multimodal traffic from users, ending with multimodal traffic at a level 35% higher than the initial multimodal traffic levels.

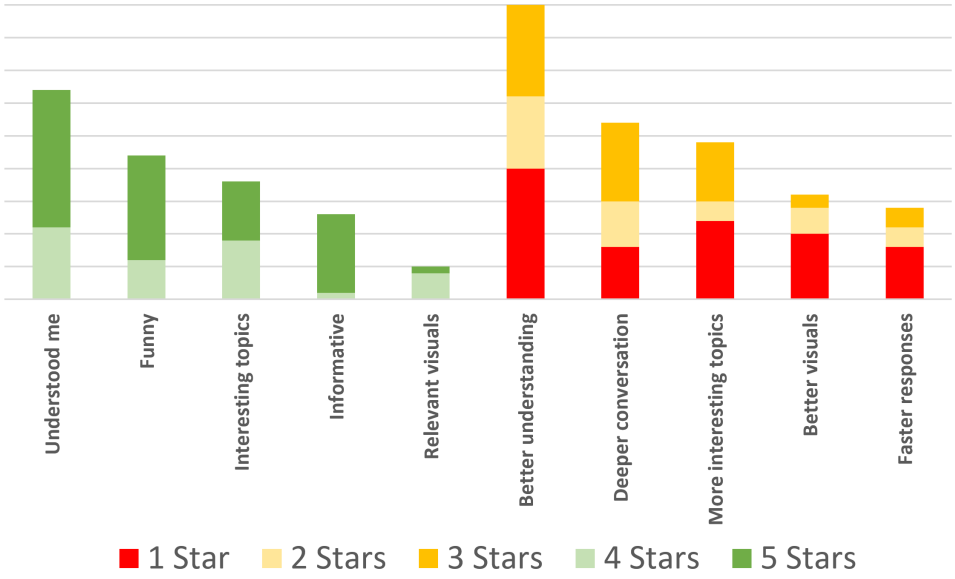

Figure 11: Positive and Negative Multimodal Feedback

The multimodal feedback flow, described in Section 2.1.3 Multimodal Experience, introduced after the end of the Semifinals period also served to increase the number of ratings and feedback provided to the teams. Figure 11 shows the relative strength of the positive feedback from customers (4 and 5 star ratings) and the relative strength of the negative feedback (1, 2, and 3 stars) from customers.

Customers have appreciated the bots' ability to understand the meaning of their statements and questions and appreciated the amusing responses provided. Customers have asked for an even deeper understanding of their comments, signaling a desire for even deeper conversations that cover more topics of interest.

4.4 Alternate Analysis to the Customer Satisfaction Rating

Although the average Customer Satisfaction Rating (CSAT) is the simplest method of identifying better conversation agents, it does incorporate various biases and can be very noisy. To combat biases such as non-normal distribution, heteroscedasticity, and non-ordinality, we also employed the signed test on the returning users to instead evaluate the probability of users rating one conversation agent higher than another. For example, given Agent A and Agent B, we only look at users that have rated both Agent A and Agent B at least once. If they rated either agent more than once, we look at the average rating given to each bot by that user. Then we perform a binomial test on whether the number

of users rating Agent A higher than Agent B follows a p=50% binomial distribution. This removes the ordinality issue because the same user is evaluating both agents and thus converts the non-ordinal rating into a binary value of true or false. The signed test is also non-parametric in nature, so we can disregard the shape of the rating distribution and the heteroscedasticity. We used this signed test method as one of the inputs to our evaluation of the relative performance of different bots.

5 Conclusion

The 5th SocialBot Grand Challenge has been driven by a whirlwind of new technologies, from the introduction of high capacity LLMs with emergent abilities to follow instructions to the shift from voice only to fully multimodal conversational experiences. The teams advanced both the art and the science of open-domain conversations and with each advancement came new research questions and new engineering hurdles – such as mitigating ever increasing challenges in latency.

One key finding in SGC5 is that while in previous iterations of the competition in order to remain competitive, bots had to rely on large amounts of curated content such as hand crafted flows or rule-based responders in addition to using neural generation as a backup or catch-all. For the first time in SGC5, we see bots achieving competitive performance without leveraging hand crafted content and instead using prompts to large language models in order to drive the conversation on all topics. It is important to note that user expectations have shifted, while primarily LLM based bots are highly competitive relative to other bots in SGC5, ratings are lower across the board compared to SGC4.

Another key development in SGC5 (as in the field more broadly) is that LLMs with instruction prompting and in-context learning have become the key workhorse across all bots and tasks. In the challenge they were applied to numerous different tasks including: formulating search queries for knowledge or image retrieval, creation of synthetic corpora, distillation of knowledge graphs, automated dialog evaluation, topic and intent classification, user engagement tracking, neural generation from knowledge, and dialog management for open-domain dialog.

The shift to multimodal interaction provided users with a more rich and engaging experience. There is no 'standard' way to augment open-domain conversation with images and multimedia and teams explored numerous different ideas. This included selection and generation of images and video content, graphical presentation of hints and options, and 'karaoke-style' animated display of system prompts. Some teams applied generative models to create custom graphics based on the current conversational context. This advancement led teams to explore image verification to ensure the images generated were worthwhile, meaningful, and safe. As with most of the innovations this year this also introduced additional latency, forcing teams to favor offline image generation. Other teams explored using realistic avatars which were included as background videos to augment the experience by matching the tone and the topic of the conversation.

One constant challenge across all bots has been latency. Users perceive latency between their input and the SocialBot's response as a critical factor when assigning user ratings. Fast evolving research in LLMs provided teams with a constant stream of new models to experiment with. Each team focused on a range of different applications of large language models. With every advancement though came the burden of additional latency. Teams adopted parallel processing, progressive responses, and pre-fetching likely responses to decrease perceived latency. Also, in many cases teams chose to optimize the performance of smaller LLMs in order to reduce latency and control costs.

While we are significantly closer to having deep and meaningful interactions that last longer than 20 minutes, there is still room for further innovation. Widespread awareness of the concept of open-domain chatbots, driven by public access to text-based large language models, drove up the user expectation bar for what a SocialBot should be able to do. With the addition of larger LLM-based responders, we observed that competition bots were able to engage with users on a broader range of topics than in earlier iterations of the competition. Many areas for improvement remain however including reducing hallucinations and contradictory responses. Based on observations from finals, work remains to improve topic tracking and engagement monitoring as bots will still often try to change topic before the user is ready to move on. Also systems are still limited in their ability to personalize the experience to the user over multiple sessions.

The 5th SocialBot Grand Challenge has left us with an entirely new set of questions to answer. Instruction prompting of LLMs has introduced a broader and more diverse set of responses at the

cost of more latency. In many SocialBots, individual topic specific responders have been replaced with complex prompts to single large LLMs, often incorporating information retrieved from external knowledge sources. Multimodal devices have allowed teams to provide more compelling responses by intertwining visual elements. With each new advancement in open domain conversations we have the opportunity to explore broader and more fulfilling experiences for the user.

Acknowledgments

We would like to thank all the university students and their advisors (Alexa Prize SocialBot Teams) who participated in the competition. We thank Amazon leadership and Alexa principals within the Alexa Natural Understanding (NU) organization for their vision and support through this entire program; Marketing for helping drive the right messaging and traffic to the Alexa Prize skill, ensuring that the participating teams received real world feedback for their research; and Alexa Engineering for the work on enabling the Alexa Prize skill. We are grateful to the Alexa Developer Experience and Customer Trust (ADECT) Gadgets team for the many certification requests they worked on quickly to certify the university bots. We would also like to thank the NU-Customer Experience team for exemplifying customer obsession by providing teams with critical inputs on building the best user experiences. We thank our leaders who took the time to virtually visit the university teams, learning from the teams and probing them to help them improve their designs. The competition would not have been possible without the support of all Alexa organizations including Speech, NU, and Data Services. We would like thank our judges and interactors for their time and expertise in the making of the finals event and Science Prize judging. And finally, we would like to thank Alexa users who engaged in many interactions experiencing the Alexa Prize SocialBots and providing feedback that helped teams improve over the course of the year.

References

[1] Apl for screen devices reference, 2023. https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-for-screen-devices.html.

[2] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, E. Goffinet, D. Heslow, J. Launay, Q. Malartic, B. Noune, B. Pannier, and G. Penedo. Falcon-40B: an open large language model with state-of-the-art performance. 2023. URL https://huggingface.co/tiiuae/falcon-40b.

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

[4] R. A. Chi, J. Kim, S. Hickmann, S. Li, G. Chi, T. Atchariyachanvanit, K. Yu, N. A. Chi, G. Dai, S. Rammoorthy, J. H. Wang, P. Sarthi, V. Adams, B. Y. Xu, B. Z. Xu, K. Park, S. Cao, , and C. D. Manning. Dialogue distillery: Crafting interpolable, interpretable, and introspectable dialogue from llms. In *Alexa Prize SocialBot Grand Challenge 5 Proceedings*, 2023. URL https://www.amazon.science/publications/dialogue-distillery-crafting-interpolable-interpretable-and-introspectable-dialogue-from-llms.

[5] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

[6] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei. Scaling instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

[7] Z. Dai, A. T. Chaganty, V. Zhao, A. Amini, Q. M. Rashid, M. Green, and K. Guu. Dialog inpainting: Turning documents into dialogs. abs/2205.09073, 2022. URL https://api.semanticscholar.org/CorpusID:248863311.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[9] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=r1l73iRqKm.

[10] P. Ehlen and M. Johnston. Speak4it: Multimodal interaction for local search. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, ICMI-MLMI '10, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450304146. doi: 10.1145/1891903.1891917. URL https://doi.org/10.1145/1891903.1891917.

[11] M. Estecha-Garitagoitia, M. Rodríguez-Cantelar, A. G. Ruiz, C. G. F. García, S. E. Romero, C. Conforto, A. S. Fernández, L. F. F. Salvador, and L. F. D'Haro. Thaurus: An innovative multimodal chatbot based on the next generation of conversational ai. In *Alexa Prize SocialBot Grand Challenge 5 Proceedings*, 2023. URL https://www.amazon.science/publications/thaurus-an-innovative-multimodal-chatbot-based-on-the-next-generation-of-conversational-ai.

[12] J. Fainberg, B. Krause, M. S. Dobre, M. Damonte, E. Kahembwe, D. Duma, B. L. Webber, and F. Fancellu. Talking to myself: self-dialogues as data for conversational agents. *ArXiv*, abs/1809.06641, 2018. URL https://api.semanticscholar.org/CorpusID:52292999.

[13] Y. Fan, K. K. Bowden, W. Cui, W. Chen, V. Harrison, A. Ramirez, S. Agashe, X. G. Liu, N. Pullabhotla, N. Q. J. Bheemanpally, S. Garg, M. Walker, , and X. E. Wang. Athena 3.0: Personalized multimodal chatbot with neuro-symbolic dialogue generators. In *Alexa Prize SocialBot Grand Challenge 5 Proceedings*, 2023. URL https://www.amazon.science/publications/athena-3-0-personalized-multimodal-chatbot-with-neuro-symbolic-dialogue-generators.

[14] R. Gabriel, Y. Liu, A. Gottardi, M. Eric, A. Khatri, A. Chadha, Q. Chen, B. Hedayatnia, P. Rajan, A. Binici, et al. Further advances in open domain dialog systems in the third alexa prize socialbot grand challenge. 2019. URL https://www.amazon.science/publications/further-advances-in-open-domain-dialog-systems-in-the-third-alexa-prize-socialbot-grand-challenge.

[15] K. Gopalakrishnan, B. Hedayatnia, Q. Chen, A. Gottardi, S. Kwatra, A. Venkatesh, R. Gabriel, and D. Hakkani-Tür. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pages 1891–1895, 2019. doi: 10.21437/Interspeech.2019-3079. URL http://dx.doi.org/10.21437/Interspeech.2019-3079.

[16] A. Gottardi, O. Ipek, G. Castellucci, S. Hu, L. Vaz, Y. Lu, A. Khatri, A. Chadha, D. Zhang, S. Sahai, et al. Alexa, let's work together: Introducing the first alexa prize taskbot challenge on conversational task assistance. 2022. URL https://www.amazon.science/publications/alexa-lets-work-together-introducing-the-first-alexa-prize-taskbot-challenge-on-conversational-task-assistance.

[17] M. Grinberg. *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.", 2018.

[18] P. Gupta, C. Jiao, Y.-T. Yeh, S. Mehri, M. Eskenazi, and J. Bigham. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.33. URL https://aclanthology.org/2022.emnlp-main.33.

[19] B. Hedayatnia, K. Gopalakrishnan, S. Kim, Y. Liu, M. Eric, and D. Hakkani-Tur. Policy-driven neural response generation for knowledge-grounded dialog systems. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 412–421, 2020. URL https://arxiv.org/abs/2005.12529.

[20] B. Hedayatnia, D. Jin, Y. Liu, and D. Hakkani-Tur. A systematic evaluation of response selection for open domain dialogue. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 298–311, Edinburgh, UK, Sept. 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.sigdial-1.30.

[21] S. Hu, Y. Liu, A. Gottardi, B. Hedayatnia, A. Khatri, A. Chadha, Q. Chen, P. Rajan, A. Binici, V. Somani, et al. Further advances in open domain dialog systems in the fourth alexa prize socialbot grand challenge. In *Alexa Prize SocialBot Grand Challenge 4 Proceedings*, 2021. URL https://www.amazon.science/publications/further-advances-in-open-domain-dialog-systems-in-the-fourth-alexa-prize-socialbot-grand-challenge.

[22] D. Jin, S. Liu, Y. Liu, and D. Hakkani-Tur. Improving bot response contradiction detection via utterance rewriting. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 605–614, Edinburgh, UK, Sept. 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.sigdial-1.56.

[23] C. Khatri, B. Hedayatnia, A. Venkatesh, J. Nunn, Y. Pan, Q. Liu, H. Song, A. Gottardi, S. Kwatra, S. Pancholi, M. Cheng, Q. Chen, L. Stubell, K. Gopalakrishnan, K. Bland, R. Gabriel, A. Mandal, D. Hakkani-Tür, G. Hwang, N. Michel, E. King, and R. Prasad. Advancing the state of the art in open domain dialog systems through the alexa prize. *2nd Proceedings of the Alexa Prize*, 2018. URL https://www.amazon.science/publications/advancing-the-state-of-the-art-in-open-domain-dialog-systems-through-the-alexa-prize.

[24] H. Kim, J. Hessel, L. Jiang, X. Lu, Y. Yu, P. Zhou, R. L. Bras, M. Alikhani, G. Kim, M. Sap, and Y. Choi. Soda: Million-scale dialogue distillation with social commonsense contextualization, 2022. URL https://arxiv.org/abs/2212.10465.

[25] O. Kobza, J. Čuhel, T. Gargiani, D. Herel, and P. Marek. Alquist 5.0: Dialogue trees meet generative models. a novel approach for enhancing socialbot conversations. In *Alexa Prize SocialBot Grand Challenge 5 Proceedings*, 2023. URL https://www.amazon.science/publications/alquist-5-0-dialogue-trees-meet-generative-models-a-novel-approach-for-enhancing-socialbot-conversations.

[26] M. Komeili, K. Shuster, and J. Weston. Internet-augmented dialogue generation. In *Annual Meeting of the Association for Computational Linguistics*, 2021. URL https://api.semanticscholar.org/CorpusID:236034557.

[27] B. Krause, M. Damonte, M. Dobre, D. Duma, J. Fainberg, F. Fancellu, E. Kahembwe, J. Cheng, and B. Webber. Edina: Building an open domain socialbot with self-dialogues. 09 2017. URL https://sorinmd.github.io/publication/edina.

[28] A. Kumar, A. Gupta, J. Chan, S. Tucker, B. Hoffmeister, M. Dreyer, C. Monson, and A. Kumar. Just ask: Building an architecture for extensible self-service spoken language understanding. *ArXiv*, abs/1711.00549, 2017. URL https://arxiv.org/abs/1711.00549.

[29] C. P. Le, L. Dai, M. Johnston, Y. Liu, M. Walker, and R. Ghanadan. Improving open-domain dialogue evaluation with a causal inference model. In *Proceedings of the International Workshop on Spoken Dialog Systems (IWSDS)*, 2023. URL https://arxiv.org/abs/2301.13372.

[30] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. 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, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

[31] L. Li, Z. Liu, L.-W. Chen, T.-C. Chi, and A. I. Rudnicky. Tartan: an llm driven socialbot. In *Alexa Prize SocialBot Grand Challenge 5 Proceedings*, 2023. URL https://www.amazon.science/publications/tartan-an-llm-driven-socialbot.

[32] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu. 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, Nov. 2017. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I17-1099.

[33] J. L. Lins, N. Reddy, A. R. Khan, M. Kowsher, A. Gusain, Y. Reddy, X. Tang, P. Jhanglani, N. Zahan, M. Zhang, Y. Yu, D. Shah, and J. Xu. From hybrid dialogers to neural responders. In *Alexa Prize SocialBot Grand Challenge 5 Proceedings*, 2023. URL https://www.amazon.science/publications/from-hybrid-dialogers-to-neural-responders.

[34] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019. URL https://api.semanticscholar.org/CorpusID:198953378.

[35] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment, 2023. URL https://arxiv.org/pdf/2303.16634.pdf.

[36] S. Mehri and M. Eskenazi. Unsupervised evaluation of interactive dialog with DialoGPT. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 225–235, 1st virtual meeting, July 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.sigdial-1.28.

[37] S. Mehri, Y. Feng, C. Gordon, S. H. Alavi, D. Traum, and M. Eskenazi. Interactive evaluation of dialog track at DSTC9. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5731–5738, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.616.

[38] Y. Nie, M. Williamson, M. Bansal, D. Kiela, and J. Weston. 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, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.134. URL https://aclanthology.org/2021.acl-long.134.

[39] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022. URL http://arxiv.org/abs/2203.02155. cite arxiv:2203.02155.

[40] O. Patil, L. Reed, K. K. Bowden, J. Juraska, W. Cui, V. Harrison, R. Rajasekaran, A. Ramirez, C. Li, E. Zamora, P. Lee, J. Bheemanpally, R. Pandey, A. Ratnaparkhi, and M. Walker. Athena 2.0: Discourse and user modeling in open-domain dialog. In *Alexa Prize SocialBot Grand Challenge 4 Proceedings*, 2020. URL https://www.amazon.science/alexa-prize/proceedings/athena-2-0-discourse-and-user-modeling-in-open-domain-dialogue.

[41] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

[42] H. Rashkin, E. M. Smith, M. Li, and Y.-L. Boureau. 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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1534. URL https://aclanthology.org/P19-1534.

[43] R. G. Reddy, S. Chandra, M. S. Sidhu, H. Bai, W. Yao, P. Pillai, K. Aggarwal, L. Ren, P. Sonawane, K. Han, V. Goyal, S. Agrawal, and C. Zhai. Charmbana: Progressive responses with real-time internet search for knowledge-powered conversations. In *Alexa Prize SocialBot Grand Challenge 5 Proceedings*, 2023. URL https://www.amazon.science/publications/charmbana-progressive-responses-with-real-time-internet-search-for-knowledge-powered-conversations.

[44] S. Saha, S. Das, E. Soper, E. Pacquetet, and R. K. Srihari. Proto: A neural cocktail for generating appealing conversations. In *Alexa Prize SocialBot Grand Challenge 4 Proceedings*, 2021. URL https://www.amazon.science/alexa-prize/proceedings/proto-a-neural-cocktail-for-generating-appealing-conversations.

[45] Y. Shen, J. Qi, S. Wang, B. M. Yao, M. Liu, Z. Xu, T. Ashby, and L. Huang. Hokiebot: Towards personalized open-domain chatbot with long-term dialogue management and customizable automatic evaluation. In *Alexa Prize SocialBot Grand Challenge 5 Proceedings*, 2023. URL https://www.amazon.science/publications/hokiebot-towards-personalized-open-domain-chatbot-with-long-term-dialogue-management-and-customizable-automatic-evaluation.

[46] H. Shi, L. Ball, G. Thattai, D. Zhang, L. Hu, Q. Q. Gao, S. Shakiah, X. Gao, A. Padmakumar, B. Yang, C. Chung, D. Guthy, G. Sukhatme, K. Arumugam, M. Wen, O. Ipek, P. Lange, R. Khanna, S. Pansare, V. Sharma, C. Zhang, C. Flagg, D. Pressel, L. Vaz, L. Dai, P. Goyal, S. Sahai, S. Liu, Y. Lu, A. Gottardi, S. Hu, Y. Liu, D. Hakkani-Tür, K. Bland, H. Rocker, J. Jeun, Y. Rao, M. Johnston, A. Iyengar, A. Mandal, P. Natarajan, and R. Ghanadan. Alexa, play with robot: Introducing the first alexa prize simbot challenge on embodied ai. In *Alexa Prize SimBot Challenge Proceedings*, 2023. URL https://www.amazon.science/alexa-prize/proceedings/alexa-play-with-robot-introducing-the-first-alexa-prize-simbot-challenge-on-embodied-ai.

[47] K. Shuster, J. Xu, M. Komeili, D. Ju, E. M. Smith, S. Roller, M. Ung, M. Chen, K. Arora, J. Lane, M. Behrooz, W. Ngan, S. Poff, N. Goyal, A. Szlam, Y.-L. Boureau, M. Kambadur, and J. Weston. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *ArXiv*, abs/2208.03188, 2022. URL https://arxiv.org/abs/2208.03188.

[48] E. M. Smith, M. Williamson, K. Shuster, J. Weston, and Y.-L. Boureau. 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, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.183. URL https://aclanthology.org/2020.acl-main.183.

[49] S. Soltan, S. Ananthakrishnan, J. FitzGerald, R. Gupta, W. Hamza, H. Khan, C. Peris, S. Rawls, A. Rosenbaum, A. Rumshisky, C. S. Prakash, M. Sridhar, F. Triefenbach, A. Verma, G. Tur, and P. Natarajan. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model, 2022. URL https://arxiv.org/abs/2208.01448.

[50] W. Wahlster. *SmartKom: Foundations of Multimodal Dialogue Systems (Cognitive Technologies)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 3540237321.

[51] H. Wang, W. Wang, R. Saini, M. Zhukova, and X. Yan. Gauchochat: Towards proactive, controllable, and personalized social conversation. In *Alexa Prize SocialBot Grand Challenge 5 Proceedings*, 2023. URL https://www.amazon.science/publications/gauchochat-towards-proactive-controllable-and-personalized-social-conversation.

[52] X. Wang, W. Shi, R. Kim, Y. J. Oh, S. Yang, J. Zhang, and Z. Yu. Persuasion for good: Towards a personalized persuasive dialogue system for social good. pages 5635–5649, 01 2019. doi: 10.18653/v1/P19-1566. URL https://aclanthology.org/P19-1566/.

[53] J. Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, jan 1966. ISSN 0001-0782. doi: 10.1145/365153.365168. URL https://doi.org/10.1145/365153.365168.

[54] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-Art Natural Language Processing. pages 38–45. Association for Computational Linguistics, Oct. 2020. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[55] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan. 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, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.30. URL https://aclanthology.org/2020.acl-demos.30.

[56] M. Zhong, Y. Liu, D. Yin, Y. Mao, Y. Jiao, P. Liu, C. Zhu, H. Ji, and J. Han. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.131. URL https://aclanthology.org/2022.emnlp-main.131.

[57] N. Zhou and D. Jurgens. Condolence and empathy in online communities. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 609–626, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.45. URL https://aclanthology.org/2020.emnlp-main.45.

[58] P. Zhou, K. Gopalakrishnan, B. Hedayatnia, S. Kim, J. Pujara, X. Ren, Y. Liu, and D. Hakkani-Tur. 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*, Singapore and Online, 2021. Association for Computational Linguistics. URL https://arxiv.org/abs/2109.06427.