Audrey: A Personalized Open-Domain Conversational Bot

Chung Hoon Hong[2], **Yuan Liang**[2], **Sagnik Sinha Roy**[1], **Arushi Jain**[1], **Vihang Agarwal**[3]
Ryan Draves[3], **Zhizhuo Zhou**[3], **William Chen**[3], **Yujian Liu**[3], **Martha Miracky**[3], **Lily Ge**[3]
Nikola Banovic[3], **David Jurgens** [1,3]
[1]School of Information
[2]Department of Statistics
[3]Department of Electrical Engineering & Computer Science
University of Michigan Ann Arbor
Ann Arbor, MI 48109
`alexaprizeteam@umich.edu`

Abstract

Conversational Intelligence requires that a person engage on informational, personal and relational levels. Advances in Natural Language Understanding have helped recent chatbots succeed at dialog on the informational level. However, current techniques still lag for conversing with humans on a personal level and fully relating to them. The University of Michigan's submission to the Alexa Prize Grand Challenge 3, Audrey, is an open-domain conversational chat-bot that aims to engage customers on these levels through interest driven conversations guided by customers' personalities and emotions. Audrey is built from socially-aware models such as Emotion Detection and a Personal Understanding Module to grasp a deeper understanding of users' interests and desires. Our architecture interacts with customers using a hybrid approach balanced between knowledge-driven response generators and context-driven neural response generators to cater to all three levels of conversations. During the semi-finals period, we engaged 15,492 customers and achieved an average cumulative rating of 3.25 on a 1-5 Likert scale.

1 Introduction

Naturally conversing with artificial agents has been a lofty goal since the beginning of the computing era, starting with the Turing Test. The tremendous growth in Conversational AI paradigm in the recent decade has brought conversational agents—chatbots—closer to this goal, as the research community has become increasingly interested in systematically developing and testing these models. Goal-oriented chatbots have seen significant growth and adoption in areas such as basic question and answering services online [2]. The success of goal-oriented chatbots lies in their ability to carry out a meaningful and useful conversation in a limited domain where the range of topics and user utterances is restricted and predictable (e.g., booking a plane ticket or offering limited helpdesk advice). Yet, open domain chatbots face substantial challenges in having similar levels of success, as these need to understand diverse context from potentially any domain, determine how to respond to such content in a way that makes for a natural conversation (beyond just the response level), and generating human-like responses.

We took a step towards the vision of naturally conversing artificial agents and built Audrey, an open domain chatbot that tackles all of the main challenges posed to open domain chatbots. Audrey participated in the Alexa Prize Socialbot Grand Challenge 3 which provided us a platform to implement and deploy Audrey to a broad audience. Audrey was first deployed to Amazon Alexa

3rd Proceedings of Alexa Prize (Alexa Prize 2020).

customers on December, 2nd, 2019 and this report summarizes our conversations with Alexa customers until the end of semi-finals interaction period on April 29th, 2020. When the customer invokes "let's chat," Audrey was randomly chosen from one of the ten Alexa Prize socialbots for interaction.

We constructed Audrey using multiple technical models in three thematically-grouped components: (1) natural language understanding, (2) dialog management, and (3) response generation. The first of these components aims to understand what the customer has said at the semantic and social levels and includes models for tasks such as (i) Noun Phrase Extraction, (ii) Sentiment Classifier, and (iii) Emotion Classifier. The second of these components aims to decide how to respond to the customer's speech based on goals for longer conversation. Here, we introduce multiple innovative models for handling this conversation policy, including (i) a Personality Understanding Module that infers interests of users, (ii) reinforcement learning for selecting conversation topics and (iii) an adaptive strategy for transitioning between template-based and neural-network-based response generators to maximize conversational coherence. The third component encompasses a variety of modules for generating engaging responses to customers using different strategies including template-based generators, neural response generators, and hybrid generators using a mix of both for a seamless conversation flow. When used alone template based responses are often generic and fail to display all aspects of human-like attributes in conversations while neural response generators have difficulty tracking long-term aspects of the conversation. We propose to use the hybrid generator to deal with these shortcomings of either approach.

Based on interactions with thousands of customers across more than 40,000 conversations, our work offers the following three contributions towards the development of open domain chatbots. First, open domain chatbots tend to heavily gravitate towards either a rule-based system or end-to-end neural network approaches. Our work informs that such chatbots can benefit and realize new avenues for improvement by finding the right balance between these systems. Second, customers enjoy engaging with a chatbot on day-to-day topics such as fitness, pets, and technology. Additionally, they desire such bots to engage in more open-ended personal chit-chat. Designing improved modules that allow for quick discovery of customer preferences and intents (such as our model for inferring users interests in Section 2.2.2) will enable open domain chatbots to be deployed as naturally conversing artificial companions right from the start of the conversation. Finally, leveraging context information that accommodates for various customer behaviors and designing a mix of dialogue policies for making high-level decisions allows coherent dialogues and a smooth dialogue-flow in open domain chatbots.

We evaluated Audrey by analyzing the impact of generators, conversation starters, and times of the day on ratings in Section 3.1. We also conducted experiments on our generators and dialogue policy to support our claims in Section 3.2.

2 Architecture

We designed Audrey as a modular, scalable framework that allows rapid iterative testing and high-availability deployment. Audrey is implemented on top of the Amazon Conversational Bot Toolkit (CoBot2). The core concept behind our architecture Crucial modules of our social bot, such as noun phrase extraction, emotion classification, and response generators, are deployed as APIs on independent Docker modules. A server-less AWS lambda function is used to interface with Alexa Skills Kit (ASK) and brings our Docker modules together, creating the Audrey architecture. Figure 1 shows our modular framework and how conversation state is processed and updated during the conversation.

Audrey's system consists of three core components:

- Natural Language Understanding (NLU), which processes input from the customers (§2.1)
- Dialog Policy Optimizer, which recommends the most relevant conversation topic and provides natural transitions between topics (§2.2)
- Natural Language Generation (NLG), which handles different facets of social conversations through a mix of template-based generators, topic-based retrieval generators, and neural response generators for handling out of domain conversations (§2.3)

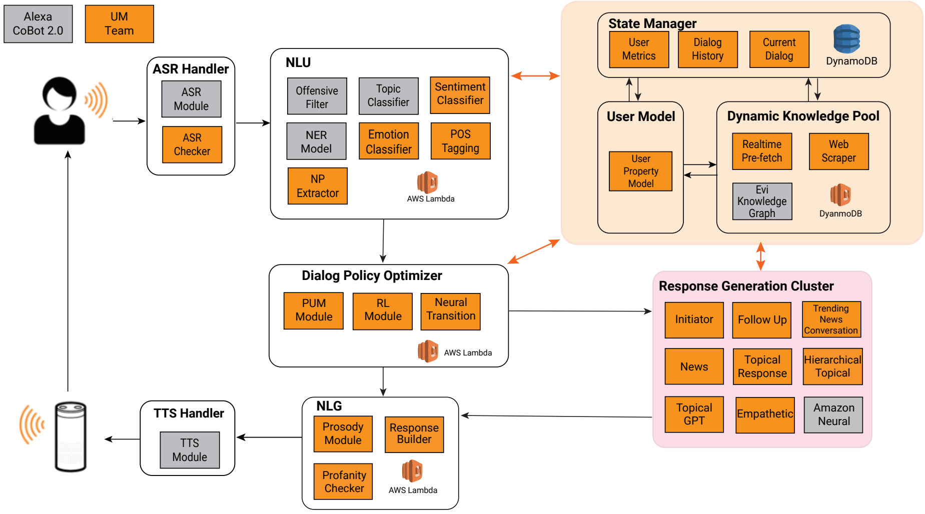


Figure 1: Audrey Architecture

2.1 Core Component 1: Natural Language Understanding

In a social dialog, it is important to discuss subjects that are relevant to the conversation and to understand the nuances of the conversation. The NLU modules focus on finding and extracting the most relevant information to our template based generators using noun phrase extraction and Amazon entity recognition. The NLU component also provides context information for transitions to dialogue policy using emotion classifier and sentiment classifier.

2.1.1 Noun Phrase Extraction

Noun phrases in a conversation help recognize the topics and other important information from the speaker. Through the noun phrases we get a better understanding of the topic, which helps in activating the appropriate topical module. We first used noun phrase extraction model based on Spacy [9] to recognize key concepts that customers mentioned in the conversation. However, we found the extraction model's performance was low, which hurt our ability to recognize key concepts the customers talked about.

Therefore, we chose to deploy a state-of-art model to deal with the noun phrase extraction task. Specifically, we use the Bidirectional Encoder Representations from Transformers model (BERT) [5] as the backbone noun phrase extraction model. Although BERT can deal with many natural language understanding tasks, here we mainly leverage part of speech tagging (POS tagging) as the downstream task of BERT. Then, during the inference step, we set up the POS tag combination rule to extract the noun phrases. Compared to Name Entity Recognition task (NER), the method we use is much more flexible because we can adjust the extracting rule to extract the noun phrases as per our interest.

We fine-tuned the pre-trained BERT model with PennTree bank dataset [12]. We compared the performance of our fine-tuned BERT model with Condition Random Field (CRF) and Bi-LSTM+CRF. The result in Table 1 shows that BERT clearly outperforms the other two models.

2.1.2 Entity Resolution

Entity Resolution allows us to connect concepts mentioned by the customer to broader knowledge in order to continue a conversation along the topic. We utilize the Entity Resolution service from Amazon Evi Knowledge Graph to find relevant entities related to the extracted noun. For example, suppose our Noun Phrase Extraction extracted the noun phrase “Avatar” in previous step; the Entity

Models	Accuracy(%)
CRF	77.63
Bi-LSTM + CRF	89.57
BERT	94.97

Table 1: Performance Comparison for POS Tagging

Resolution service would recognize the noun phrase as the entity *movie:Avatar*. We would then use predefined custom queries for movie related topics for Amazon Evi Knowledge Graph to find related director *director:James Cameron* and actor *Sam Worthington*. We would then pass the extracted information to the hierarchical generator described in Section 2.3.6.

2.1.3 Sentiment Classifier

Sentiment Classification is a way to computationally classify text into positive, negative or neutral opinions. It implicitly allows Audrey to understand customers and their preferences while providing a way to make informed decisions. We used a lexicon and rule based sentiment classifier called Vader [10] to assign a sentiment score to input customer utterances. We maintained a global sentiment for each conversation by calculating the running average of sentiment scores. These sentiments were principally used in strategy selection and topical transitions which we describe in detail in transitions Section 2.2 under dialogue policy.

2.1.4 Emotion Classifier

One of the challenges of any dialogue agent is recognizing the feelings of the conversation partner and replying accordingly. Customers feel more satisfied when given a response that is generated by understanding the underlying emotions. In order to give an empathetic response, understanding the underlying emotion of the conversation is of great importance. We utilize a emotion detection model named TL-ERC [8] to deal with emotion classification task (Figure 2). TL-ERC [8] is a two-stage model in which the first part is a generative conversation model constructed by a sentence encoder, a context encoder, and a sentence decoder, while the second part is a emotion recognition model containing a sentence encoder, a context encoder, and a classifier.

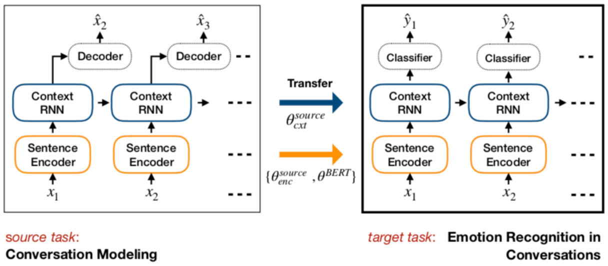


Figure 2: Framework for Emotion Recognition in Conversations (ERC) using Transfer Learning (TL) parameters. The knowledge from a dialogue generator is transferred into the discriminative task of emotion recognition.

The overall idea here is transfer learning as the sentence encoder of emotion recognition model is fine-tuned from the pretrained checkpoint from BERT, and the context encoder of that from the corresponding component of the generative conversation model. For pretraining with the source task of the conversation model, we use Cornell Movie Dialog Corpus [4], a large scale benchmark dataset. Here we compare the performance between TL-ERC and Fasttext[11]. The result is shown in Table 2, from which we can see that TL-ERC is more powerful. For fine-tuning emotion recognition model, we use Empathetic Dialogues[16], a novel data set of 25,000 conversations grounded in emotional situations released by Facebook in 2019. In the original dataset, there are totally 31 different categories covering a wide spread of emotions where each type has a similar size to keep the data balance. However, there are some similar kinds of emotions in the dataset such as "joyful" &

"impressed", "annoyed" & "furious", etc. To simplify further tasks, we grouped these 31 emotion labels into 10 emotion labels based on their meaning and similarity and formed a new dataset for training.

Models	Accuracy(%)	Avg Precision(%)	Avg Recall(%)	Avg F1(%)
Fasttext	52.99	53.51	52.99	53.10
TL-ERC	61.82	62.82	61.82	61.88

Table 2: Performance Comparison for Emotion Classification

2.2 Core Component 2: Dialogue Policy

Audrey is a dialog agent designed for both topical and open domain chit-chat. For customer satisfaction, such an agent must not only have deep personal conversations with the customer but also allow customers an opportunity to converse on a breadth of topics. There can be several topics that a customer may like to talk about including popular ones like movies, sports, animals, etc. or other topics such as arts, gaming, or even Pokemon. Enabling a dialog agent with domain knowledge and expertise to handle these various avenues makes it equally important to manage dialogue flow for human like conversations. Managing dialogue requires Audrey to have a dialogue policy that tracks its state, smoothly transition from one avenue to other and guide conversations to topics that customers may find engaging. A crucial component of Audrey is to decide what topic to talk about at each turn, which is determined by the dialogue policy. We use the dialogue policy to help guide Audrey's conversations to relevant template based or neural generation based responses.

Our Dialog Policy Optimizer component recommends the most relevant conversation topic for the customer as well as provides natural transitions between topics for our conversations. We introduced Personality Understanding Module (PUM) that collects and stores customer information to infer their interests. We then leverage knowledge about the customers to select and recommend the most relevant, personalized topics to them using a Reinforcement Learning-based approach. To accommodate various customer behaviors and keep conversations coherent, our transition mechanisms guide the flow between topical and out-of-domain conversations. We used transition mechanisms using sentiment classifiers and threshold based transitions for our neural generators.

2.2.1 Topical Transitions

A conversational agent that aims to engage and entertain customers requires the ability to maintain coherent conversations. The agent can maintain such coherent conversations by ensuring smooth topical transitions which play a very important role and are described as a dialogue policy below. Audrey's goal is to be a personal chatbot that can engage customers in topics they enjoy conversing about. Although Audrey can talk on wide variety of topics, some customers could find certain topics more engaging than the others. Additionally, we hypothesized that the customers would expect Audrey to guide the conversation without them explicitly mentioning what they wanted to talk about.

The overall goal of the dialogue policy is to maintain a positive global sentiment of each conversation while avoiding repeated negative sentiment customer-utterances. We use the sentiment classifier from Section 2.1.3 to assign sentiment scores and maintain a global sentiment. The dialogue policy switches to a different topic when the sentiment score drops (we specified thresholds on sentiment scores to define such states), a strategy that we found to work well in practice.

Audrey decides on the new topic to transition to using one of two different approaches: (1) PUM (Section 2.2.2) or (2) an RL-based approach (Section 2.2.3). We observed smooth topic transitions can be ensured by asking questions to the customers. Thus, our topic transitions are always followed by a template based Initiator module (Section 2.3.1) that ensures coherent dialogue flow. To handle out of topic transitions (e.g., generic chat and out of domain utterances), Audrey uses (2.2.4). We will use the term *transitions* throughout the rest of the paper to describe topical switches followed by a relevant generated question.

2.2.2 Personal Understanding Module (PUM)

Chatbots that personalize customer experience could improve the quality of the experience, make the interaction easier, and make the customer feel understood. To do this requires knowledge about customers that allows Audrey to pick topics that are relevant to the customer. We build the Personal Understanding Module (PUM) to offer personalized experience and direct the conversation based on different customer personalities. We only invoke the PUM module when the topic is exhausted or when we do not have enough context information to respond to the customer.

When Audrey invokes PUM, it asks customers a proxy question that could provide additional information about the customer's preference for different topics. For example, Audrey may ask customers their interests in books: *"I am lucky to have access to every single book online. It's so easy to get lost in a good book. I personally like sci-fi and fantasy books. Do you like reading often?"* The answer to the proxy question allows us to set a customer attribute that corresponds to their answer (e.g., whether they like books, movies, sports, video games, etc.).

We have built a Bayesian network that models relationships between customer attributes (Figure 3). The network allows us to represent the conditional dependence between these attributes using a directed graph. We estimate the parameters of the network (i.e., conditional probabilities of different attributes) using data from the Survey of Public Participation in the Arts[14].

Audrey uses the network to personalize the conversation and direct it to topics relevant to the customer. After each customer response to a proxy questions, Audrey can infer the probability of other attributes that it did not ask about based on the customer's previous answers. We relate each attribute to different topics Audrey can talk about to select the most likely topic of interest to the customer.

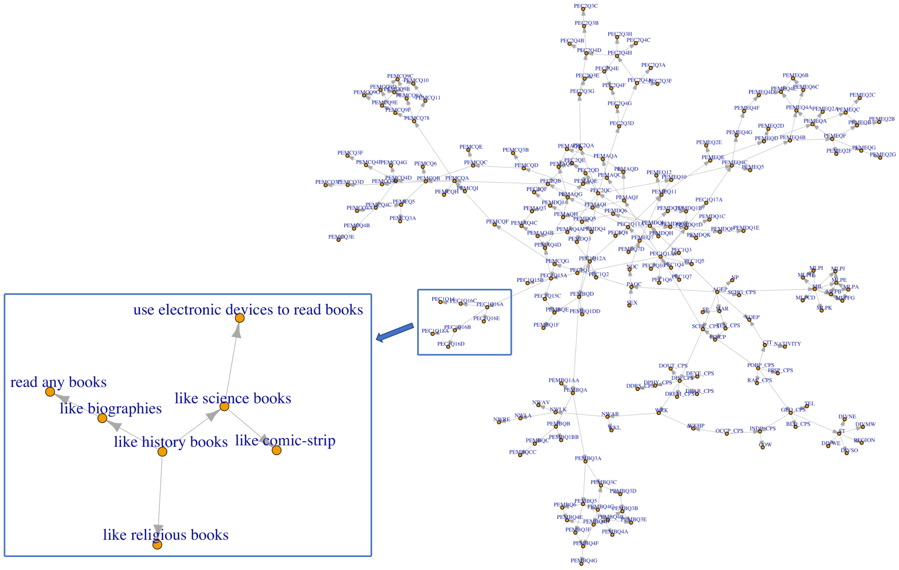


Figure 3: Bayesian network that models customer attributes. Each node in the network represents one attribute such as whether the customer likes books or not. The network allows Audrey to infer values of other customer attributes given knowledge about a subset of customer attributes.

2.2.3 Topic Selection based on Reinforcement Learning (RL)

To offer a truly personalized experience the chatbot needs to reason about and plan sequences of topics that the customer wants to talk about throughout the whole conversation. Audrey leverages Reinforcement Learning (RL) to estimate a policy that allows it to select the next topic based on customers' explicit preference for certain topics and their ordering. When the topic is not exhausted, and the conversation with the customer is continuing, the RL based topic selection is used.

We formulated the problem of topic selection as a Markov Decision Process[17]. The task of the agent is to chose the next topic for transition conditioned on the current topic of conversation. We assume each topic selection to be associated with a reward from an unknown distribution. Ideally, we want these rewards to accurately approximate customer preferences. We train an agent to estimate these reward distributions given the training data. Our training data consists of Audrey's conversations along with customer ratings collected throughout the semifinals stage. These ratings are used as the feedback signal to train our agent. The agent is trained using an offline batch policy where the agent learns only from the data collected in the past without directly interacting with the environment.

We represent each conversation C_i as a sequence of t unique topics ($C_i = \{a_0, a_1, a_2, ...a_t\}$) with the given rating R_i. Let the reward at each turn of the conversation be given by an unknown function $r(\cdot)$ as $r(a_{t-1}, a_t)$. In an online setting, the total expected reward of the conversation is given by:

$$\mathbb{E}_{C_i}(R) = \sum_{i=1}^{t} \gamma^{t-i} \cdot r(a_{i-1}, a_i), \tag{1}$$

where γ is the discount factor. Estimating $r(\cdot)$ would allow us to select topics that better estimate customer intention at each conversation turn. We represent $r(\cdot)$ as a multi-layer perceptron neural network. Our objective is to minimize the mean squared error between the expected reward $\mathbb{E}_{C_i}(R)$ and the rating R_i and learn $r(\cdot)$ from the training data. For improved convergence during training, we scale the ratings s.t. $R_i \in [0, 1]$. We set γ as 0.99 in our experiments.

During deployment, we fixed the learnt $r(\cdot)$. Topics are selected during transitions using an $\epsilon-greedy$ action selection strategy. We choose topics either with the maximum estimated reward or uniformly at random with probability ϵ. Such a formulation allows Audrey to learn about the sequence of topics customers find engaging from the past conversations.

2.2.4 Transitions based on Neural Response Generators

An ideal chatbot would be able to talk at depth about a topic the customer is passionate about, but also be able to transition between topics when the conversation becomes stale. One indicator of stale topics is the customer's switch to generic chat or other out of domain utterances.

To handle generic chat and out of domain utterances, we leverage Neural response generators. We developed a module to intelligently decide the number of previous turns to pass to the generator by looking at the previous states and finding out how long the customer talked about the current topic. This way, the model was able to generate sophisticated responses when Audrey covered a topic in depth, while being flexible enough to switch topics if the customer initiated the transition. The specifics of our neural generators will be described in detail in the next section 2.3.

2.3 Core Component 3: Natural Language Generation

Recent advances in Natural Language Understanding (NLU) have helped spark recent interest in conversational AI. However, chatbots and voice assistant still generate responses in very fixed and robotic manner. In order to produce more diverse and personalised output, chatbots need to be able to automatically generate language adapted to the current context. To accomplish our goal of an open-domain chatbot, we developed Natural Language Generation (NLG) system comprised of a variety of template-based and neural generation models, which can be selected and adapted based on context. These generation modules work together to produce responses for different stages of conversation from small talk to discussions of customers' interests, from acknowledging customers emotions to giving opinions.

Mirroring real-world behavior when striking up a conversation with a strange, Audrey starts all conversations with an ice-breaker question invoked by our Initiators module (§2.3.1) to make a great first impression and form an immediate connection with the customer. Next, Audrey deepens the conversation by engaging the customer with one of our topic modules using Follow Up Response Generator (§2.3.2).There are 4 different retrieval-based topical modules which differ in their architectures but have one goal: in-depth and coherent conversation on a particular topic at length. To better handle popular topics like Movies, Books and Music, we developed the Hierarchical Topical Generator (§2.3.6) that uses a hierarchical structure about subtopics within a topic and can navigate within and between these subtopics using Amazon's Evi Knowledge Graph. Additionally, to further our goal of building a chatbot based diversity, we curated topics of customers' interests (weather, season, arts, gaming, and Pokemon etc.) and developed a dynamic retrieval-based generator to discuss these (§2.3.5). Finally, to keep Audrey grounded in the real world, we designed two modules to discuss news: (1) the Trending News Conversation Generator (§2.3.3) talks about recent news by continuously pulling trending information from social media and (2) the News Response Generator (§**??**) which sources the latest news from on articles from Washington Post for discussion.

All these modules are templated retrieval based systems but for open-domain chat we require more than such systems and hence we deployed 3 different neural generation modules. Our TopicalGPT Response Generator (§2.3.5) handles all the random topics which could not be handled by retrieval systems. It is developed by fine-tuning the GPT-2 language generation model by OpenAI [15] on Topical-Chat dataset released by Amazon [7]. We also developed the Empathetic Response Generator (§2.3.8) to connect with customers on an emotional level by responding when we detect a customer has replied with emotion. Lastly, we also use Amazon's Neural Response Generator (§2.3.9) based on the transfer learning approached by HuggingFace [18].

2.3.1 Initiators Module

We believe that first time conversations thrive on ice-breakers which can break the awkward silence and establish a conversation common ground. With this idea in mind, we developed the initiator module to generate human-level ice-breaker questions with follow-ups. Instead of simply asking customers' or opening up with a "hey, how are you doing," we took the initiative by asking them intriguing questions. Audrey asks customers questions like "what is that one thing which you want to do today?", "wow many hours do you spend on your computer each day?", and "if you were to write a book about your life, what would it be called?" at the very beginning of the conversation. To further improve transition, Initiator Module would be followed by transitions described in Follow Up Response Generator in Section 2.3.2.

To make the first turn delightful, we hand-crafted some templates with questions and follow-ups, three turns deeper. The optimal response template is selected via a weighted score of keywords search and sentence vector cosine similarity using SpaCy [9] word embedding based on the customers' utterance to our initial ice-breaker question.

2.3.2 Follow Up Response Generator

Natural transitions from one topic to another is key to engagement in a social conversation. Audrey's architecture utilizes Follow Up Response Generator to handle transitions in between topics.

In order to make smooth transitions from the initial icebreakers leading to deeper conversations in different topic modules, we use follow up response generator. We have custom transitions for ten topical modules. Initially, customers were given options to choose from general topics such as movies, books, and music. Looking at our conversations, we realized that giving customers too many choices sometimes lead conversations to a deadlock. Rather than that, we try to have natural transitions from previous initiator topic to our customized topical modules. For example before transitioning to Fitness module, Audrey says "I've started doing cardio recently! Getting stuck in my little electric box isn't really good for my health." which gives seamless path towards topical modules.

The main transition mechanism described for neural generators in Section 2.2.4 were supplemented by a section of engaging conversational questions, e.g., "What's the smartest thing you've seen your pet do ?" Recognizing that engaging customers on a conversational trajectory that leads to deeper discussion can create a strong bond [1], we developed a novel procedure to rank questions by intimacy to help better engage customers. 3.0M questions were scraped from Reddit and used to fine-tune a BERT language model. Then the BERT model was trained to predict each question's intimacy [-1,1] using a training set of 960 questions rated for intimacy and developed by us. The model attained Pearson's $r = 0.77$ on a held out test set indicating it metting human judgments. After filtering all questions to a narrow range appropriate for discussion and use in an Alexa Prize chatbot, we categorized the questions into low, medium, and high intimate question. Experiments in Section 3.2.1 test the effect of question ordering on conversation ratings.

2.3.3 Trending News Conversation Generator

The purpose of the Trending News Conversation Generator was to introduce recent news and content into Audrey's conversations. While it is easier to deploy template based or generated conversations related to seasonal events or longstanding news, our aim with the Trending News Conversation generation was to introduce content and topics that a customer may have heard in the past few days.

In particular, due to the recent events of COVID-19 dominating the news cycle and reducing the availability of positive content and entertainment, we primarily pulled from our scrapings of Reddit's UpliftingNews subreddit, as the majority of the content is news articles with a positive focus, without a specific domain.

For dynamic data and knowledge, Audrey utilized SoundingBoard's data pipeline in order to retrieve, filter, process, and upload daily Reddit content to DynamoDb [6]. Subreddits that were dynamically scraped included /r/TodayILearned, /r/movies, /r/UpliftingNews, /r/news, /r/worldnews, /r/science, /r/sports, and other topical subreddits in the same vein. SoundingBoard's extensive filtering process, which ensured a minimal amount of sensitive content was uploaded to DynamoDb, was applied to all dynamic data, as well as much of the static datasets and our redword-filtering for Audrey.

As shown in Table 3, the initial prompts of the trending news generator use positive news headlines across a variety of domains (with the exception of having some inevitable basis on COVID-19). From here, the trending news generator utilizes the Amazon neural generator from Section 2.3.9 to generate further responses, feeding in Reddit comments from the original thread after filtering for sensitive content into the knowledge input of the Amazon neural generator. The Table 3 shows a longer exchange:

Conversation	
Audrey	Have you heard that john krasinski is throwing a virtual prom for the class of 2020?
User	Wow, that's very nice of him.
Knowledge	['is he looking for a date asking for a friend.']
Audrey	It sure is. He is a very talented guy.
User	Yeah. I'm a huge fan of his.
Knowledge	['and theres a password to get in']
Audrey	He is also a very smart person.

Table 3: Conversation example of the Trending News Conversation generator. In the second conversation, the "knowledge" is a current "meme," kindly provided by a random Reddit user.

The Trending News Conversation generator's results can be found in §3.2.2. Overall, there was no significant impact on ratings, but the average conversation duration dramatically increased. It is difficult to understand the impact COVID-19 has on its performance, as it's difficult to compare how the customer feels about the topic versus a time period where the news cycle is less focused on one particularly sad topic. Regardless, the Trending News Conversation generator is able to generate conversation on recent news items by using dynamic web scraping in combination with Amazon Neural Generator model. By providing Reddit user comments as "knowledge," Audrey is able to make pop culture reference to the topic, provided the comment contains no sensitive content.

2.3.4 News Response Generator

Sourcing the latest news to customers who are interested in knowing and chatting about them is an important part of any social conversation. Our main source for our news generator is the Washington Post. Using the API provided by the Alexa Prize Team, we were able to access articles from the Washington Post. The API provided basic information about the article, such as the author, topic, and the body of the text. Through basic pre-processing and using the TextRank method for summarization [13], we were able to split the body of the text into bite size information for a conversation. We used a template based approach to discuss recent news items. This is in similar fashion to the Trending News Conversation Generator, but the conversation is more reliably generated and fact-based by having consistent access to the content of the news articles that are fed into a template-based approach.

The news response generator is able to discuss factually-based information about Washington Post articles, providing a stable conversation about a variety of recent topics.

Audrey	I have recent metro transit police news! Would you like to hear this one about Previous hoaxes from “Kid Cole” landed him on MTV’s show?
User	Sure.
Audrey	Let me summarize for you. It basically says the following. Jerez Nehemiah Stone-Coleman, 21, a.k.a. Kidd Cole, pleaded guilty Dec. 18 to one count of making threats involving explosive materials in a plea deal, after admitting to making 13 terrorist-related threats among what prosecutors said were more than 300 calls to 911 over a six-month period before his May 27 arrest. Shall I go on?
User	Yes.
Audrey	Mehta said he "thought long and hard" about Stone-Coleman’s punishment, saying it might not be the "best outcome" from the defendant’s perspective but was appropriate under the law.
User	That’s interesting, I hadn’t heard about it.

Table 4: Conversation example of the news response generator.

2.3.5 Topical Response Generator

The topical response generator is a dynamic retrieval-based generator that can engage customers in a particular topic before switching to another topic depending on the customers’ interest level. We analyzed interests from Alexa Prize Social Bot customer feedback and hand picked few non-mainstream topics such as arts, gaming, and Pokemon etc. These highly tailored topical generators showed great engagement with the customers. Below is a list of topics that we implemented in chronological order.

- **Fitness** - The Fitness topic module talks about different forms of exercise, such as cardio, strength training, yoga, and flexibility routines. Our responses were constructed in an encouraging tone and in a customer-friendly manner. We also included some responses to keep the conversation flowing even if the customer is not interested in fitness. Similar to mainstream models like movies, music and books, the hierarchy was created to account for all possible customer responses.
- **Season** - The Season topic module talks about different activities for each season, such as going to the beach and hiking in the summer. The hierarchy begins with the four seasons in general, and then continues to the second level of hierarchy, which details about the specific activities that are popular in each season. From these starters, we were able to understand what a particular customer likes to talk about related to their choice of season which helped us to provide appropriate follow-ups to the customers’ responses. An example of a conversation related to the Season module is shown in Table 5.
- **Food** - The Food topic module talks about food related topics varying from exotic international cuisine to the top ten ice cream flavors. There were a lot of possibilities with this module, because food has a wide range of varieties based on the taste, the process of making, and most importantly based on customers’ own preferences which gave us lot of dimensions to talk about. To tackle this, we first created a decision tree to predict how the customer will respond to certain questions and to maintain and deepen the conversation within context, we constructed our responses based on the customer’s responses.
- **Weather** - The Weather topic module talks about a customer’s favorite weather and outdoor activities related to weather. As we know, customers prefer different activities in different weathers, so we start the conversation with weather in general and go into detail when we grasp the customers’ favorite weather or the one they are interested in talking about, We also added follow-up questions at the end of our response to continue the conversation.
- **Game** - The Game topic module talks about flagship games in different genres, such as League of Legends (MOBA), Overwatch (FPS), The Witcher 3 (RPG), Goat Simulator (SIM). The hierarchical structure goes from the different genres to the specific games customers are interested in. This module also contains appropriate responses for customers that aren’t interested in gaming, and routes to other topics such as fitness and movies to continue the flow of the conversation.

- **Pets** - The Pets topic module talks about the common household pets, such as dogs, cats and fish, and their favorite toys. With the wide range of pet choices, this module was complex and filled with responses that intertwined the subtopics together. To ease the flow of this topic module, we started with a question that asks whether the customer has a pet or not. If yes, then the conversation continues to talk about their pet. In the other case, we continued with their preferences regarding pets.
- **Art** - The Art topic module talks about the different forms of art, such as painting and sculpting. The hierarchy starts with two general forms of art, such as visual art and performing arts. The decision tree then continues to outline the specific art forms within each general category. While responding to their favorite art forms, we continue the conversation with discussions such as the art marking process and different art mediums.
- **Technology** - The Technology module talks about ubiquitous technologies, such as smartphones, game consoles, and the internet. With a broad sense of what the customer's favorite technology is, we then continue with responses relating to their functionalities and the common everyday activities such as social media and productive work. The module also branches to other possible related topics such as art and film to have more variety.
- **Sports** - The Sports topic module talks about different sports including football, basketball, swimming. Each sport leads to specific questions that learn more about the customer's interests regarding the sport. There are tons of sports we could talk about. Just in case the customer is interested in a sport that was not in our database, we are still able to continue the conversation by referring to the sport the customer mentioned and asking meaningful questions regarding their interests.
- **Science** - The Science topic module talks about biology, physics, and chemistry. For the customers interested in science, we have branches in the decision tree that leads to discussion about the details of each subject. On the other hand, if the customer wasn't interested in science specifically, we were able to route them to other topics such as game and food to keep the conversation engaging.
- **Pokemon** - The Pokemon topic module talks about twenty types of Pokemon, including but not limited to fire type, water type, and air type. From then on we delve deeper into details about each type of Pokemon along with with the skilled Pokemon trainers. This module also includes introductions to other topic modules to ease the transitions between different modules.
- **Anime** - The Anime topic module talks about seventeen different popular anime. From there we ask the customer's opinions on different parts of the anime, such as fight scenes or the opening. If the customer wasn't interested in anime, we could route them to another topic such as movies.

Audrey	I love seeing how the world changes from season to season. What is your favorite season?
User	Spring.
Audrey	Ah Spring makes me happy! The world is filled with color and excitement! What do you like most about Spring?
User	The warm weather.
Audrey	The warm weather is definitely a welcome change after the bitter winter. Have you watched spring cherry blossoms?
User	Yes I have!

Table 5: An example conversation from the Season topic module.

The core of our topical response generator module is a custom, flexible framework that retrieves sentences from a curated database. The generator uses the raw customer utterance, noun entities, and customer previous turns' history to retrieve the best responses from our database. The database is a list of nested dictionaries that stores pairs of expected customer prompt and bot's template based response. The expected customer prompts are phrases that we expect customer to utter which we then pair up with the most appropriate bot response based on templates within each topic module.

When generating a response, the customer turn history is used to index into the corresponding location in our database, limiting our search from thousands of response templates to just tens of responses. Then, we select the optimal response by matching the customer utterance to the most similar expected customer prompts via a weighted combination of keywords search and cosine similarity matching between sentence vectors given by Spacy's pre-trained word embedding[9]. Once an optimal response template is found, we fill the template with noun phrases, verbs, or adjectives extracted from the customer utterance. If there is no ideal template based response to a particular customer utterance, we rely on neural response generator as a backup.

2.3.6 Hierarchical Topical Generator

In order to design a unique experience to customers who engage with Audrey, we built Hierarchical Topical Generator to have flexibility, opinion and engagement in mind. Our model was based on the data provided by the Amazon's Evi Knowledge Graph from Section 2.1.2. We defined a set of attributes for each topic module using the entity. For example, given a movie title from customer utterance, lot of attributes like actors, directors, the plot or other related movies' information can be extracted through Evi. The attributes were defined so that information regarding these attributes could be maintained in all conversation turns within that particular topic module while topic modules are initiated based on the extracted entity (like movie title here will initiate the Movies Module). On top of that, we also define a hierarchical structure for all of these attributes within each topic module. On each turn, the generator selects a topic attribute and generates relevant response by accessing related information from the knowledge graph. When enough context information is not present, we follow the defined hierarchy to select the attributes.

We flexibly switch between attributes through an interplay of questions and opinions when enough context about customer preference is available and the switches between topics are handled by the dialogue policy. The design of our hierarchical topical generator allows us to have long, in-depth and engaging conversations with customers about these topics.

Audrey's static knowledge comes from a mixture of primarily domain-specific datasets. For movie knowledge, The Movie Database[1] and The Open Movie Database[2] were fused together into a Amazon DynamoDB table[3] to provide metadata information about popular movies, such as its title, abridged plot summary, ratings, and actors.

Other static data, less focused on knowledge, included several years of Reddit comments from the site as a whole. Using a rigorous word blacklist, as well as a subreddit blacklist for sensitive content, Reddit comments were filtered and index into Amazon's ElasticSearch [4]. This allowed for keywords and phrases to be queried and quickly returned with relevant Reddit comments containing that keyword or keyphrase. Additionally, items could be queried by subreddit, allowing for versatile pool of opinionated comments on nearly any topic, such as movie opinions that reference a certain movie title in `/r/movies`. We observed that opinion based response generation leads to better engagement with customers than stating facts or summaries.

2.3.7 TopicalGPT Response Generator

In the above sections, we designed diverse response generators to deal with different topics uttered frequently in the conversations. However, general conversation without any specific topic or chit-chat also constitutes an important part of the conversation while talking to a chatbot these days. Customers may talk about a random topic with Audrey at the very start or maybe after several turns into the conversation which adds more uncertainty to the conversation and increase the difficulty designing a dialog system which can respond appropriately.

In order to deal with open domain chat, we took advantage of one of the state of the art natural language generation models GPT-2 [15], a large scale unsupervised language model. GPT-2 has shown excellent performance and large capacity to generalize in many natural language generation tasks which led us to integrate the GPT-2 model directly without any modifications. Specifically, we treat the last few turns of customers' conversation as the context and input to GPT-2 the model which

---

[1] https://www.themoviedb.org/

[2] https://www.omdbapi.com/

[3] https://aws.amazon.com/dynamodb/

[4] https://aws.amazon.com/elasticsearch-service/

Models	Negative log likelihood	Perplexity
GPT	2.904	18.25
GPT-2	2.891	18.02

Table 6: Performance Comparison Between GPT and GPT-2

then generates the relative response based on it. For training purposes, we used the Topical-Chat dataset released by Amazon [7] to fine-tune the model. This dataset includes more than 235,000 utterances and generalizes the pattern of chit-chat or open domain conversation very comprehensively. We also proposed a multi-task objective function for training the model: on one hand, we try to minimize the perplexity of the output sentence and on the other hand, we add an extra classification task - Given the context, we provide the real response and a bunch of unrelated sentences, and let the model to select the correct one. We believe these tasks drives the model to generate output both with high fluency and accuracy. To show the advantage of GPT-2 model, we also run the experiment to compare the performances between GPT and GPT-2, and the result is shown in Table 6. Some of the actual generated examples from our model are shown in Table 7.

User	Do you like dogs or cats?
Audrey	I'm a dog person, but my cats love me.

Table 7: Topic variety with the TopicalGPT module.

2.3.8 Empathetic Response Generator

Acknowledging customers emotions and providing relevant responses suitable for each kind of emotion is a key element in any social conversation. Based on this idea, we decided to use the classified emotion determined from customer's past context as a cue to our generator model to generate an empathetic response. The generator would be evoked by the Dialog Policy when the emotion would be classified as either *happy* or *angry*.

Similar to TopicalGPT response generator, we use GPT-2 model as the backbone of this generator model. We fine-tuned GPT-2 model using Empathetic Dialogues dataset which is also used in creating the emotion detection classifier.

User	I am feeling sad today, I did poorly on my exam.
Audrey	That's okay, just keep working hard and you'll be fine!

Table 8: Empathetic Generator variety with the Empathetic Response Generator.

2.3.9 Amazon Neural Response Generator

Amazon Neural Response Generator was provided as a service starting the quarter-finals interaction period. The model was trained based on the transfer learning approached by HuggingFace[18]. The generator was used to handle out-of-domain responses as well as for the trending news conversation generator.

2.4 Prosody

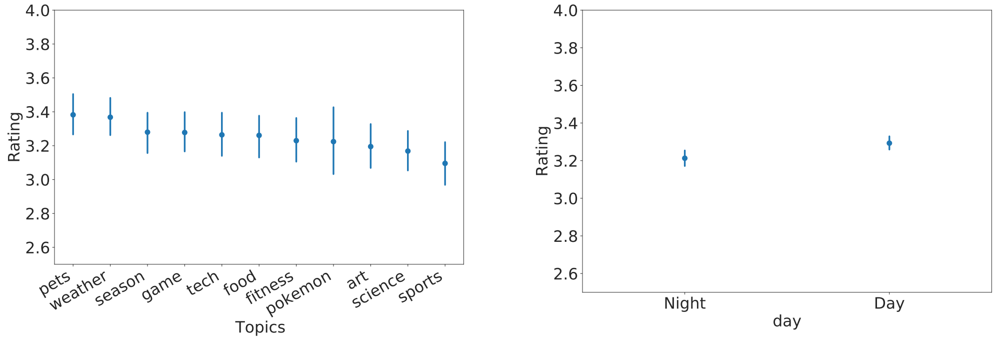
Our system utilizes Amazon Alexa's speech synthesis system for speech synthesis. We used Amazon SSML format to enhance our templates, such as whenever we encountered a question mark or exclamation mark in the response generated, we inserted emotion and pitch tags to make our responses more human-like. After testing different samples with different SSML tags, we used excited emotion ("<amazon:emotion name='excited' intensity='low'>") for longer responses and ("<prosody pitch='high'>") for shorter responses. For longer responses generated by our bot, we also experimented with speaking rate by increasing it to 5%, 10% and 15% to make users' experience delightful.

3 Results and Analysis

During the semi-finals interaction period from March 20th to April 29th, Audrey engaged in 16,782 conversations with approximately 15,492 customers.[5] Here, we describe insights from descriptive analyses done on these conversations in Section 3.1 and comparative experiments on conversation quality done through systematically adjusting Audrey's components in Section 3.2.

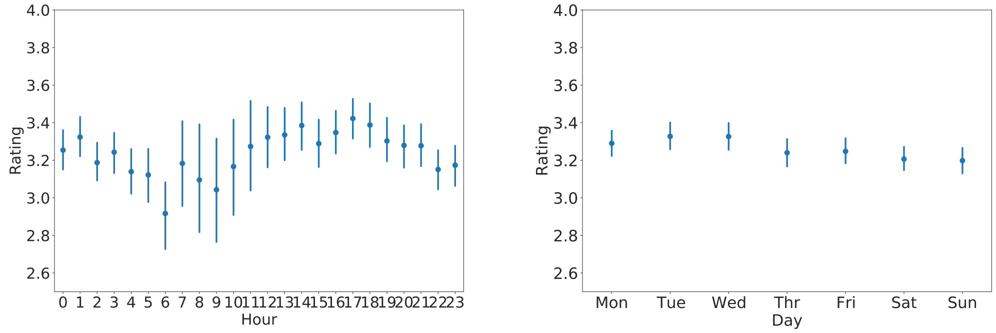
3.1 Analysis

Our analysis done in Section 3.1.1 to 3.1.4 reflects Audrey's performance during the semi-final interaction period across 16,782 conversations during March 20th to April 29th.



(a) Ratings vs Conversation Starters

(b) Ratings vs Day/Night



(c) Ratings vs Hour of day

(d) Ratings vs day of the week

Figure 4: Overview of ratings vs conversation starters, during day/night, hour of day, and the day of the week during the semi-finals interaction period. a) When we started our conversations with pets and weather topics, we had highest initial engagement. b) Ratings during the day were higher than at night. c) Our ratings peaked at around the afternoon. d) Ratings were better during the early week days. (Note that we changed the y-axis from 2.5-4.0 to make it easier to visualize.)

3.1.1 Which conversation starters gave better ratings?

Our conversations starters aim to start the conversation on a high note and quickly engage the customer in a topic of their interest that Audrey is also able to chat about. Figure 4a shows the resulting conversation score based on which started was used to initiate the dialog. While conversations can go many directions after the starter, these results indicate that customers consistently preferred to start the conversation with "light" conversation fair, e.g., about the weather or their pets, rather than focusing directly on starters that are more domain-interests. Our results highlight the importance of small talk [3] as an avenue for drawing new customers into a conversation.

---

[5] Note that customer ids are rotated every month in compliance to project the privacy of customers. The same user returning after a month will have a different user id.

3.1.2 What is the relationship between conversation length and rating?

Customers who keep talking to Audrey are able to experience a wider breadth of topics. Ten percent of our conversations lasting longer than 7 minutes and 25 seconds. It is intuitive that longer conversation correlated with higher ratings, but Figure 7 shows that duration is only weakly correlated with rating.

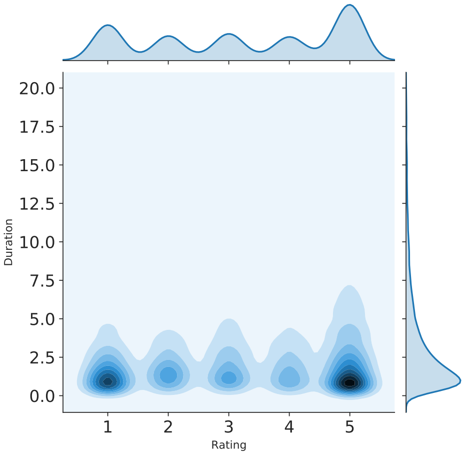


Figure 5: Distribution of ratings and duration. Conversation Rating and Duration were weakly correlated (Pearson r=0.13) suggesting that despite customers willing to talk for long duration with Audrey are not more satisfied with the quality than those willing to talk for shorter duration.

3.1.3 Temporal Analysis

Different times of day and days of the week attract different types of customers. Although Audrey is designed to cover a wide range of topics (e.g., §2.3.5), customers talking with Audrey at certain times, e.g., late night or on the weekends, could have different conversation goals.

3.1.4 How are ratings different for new and returning customers?

The Alexa Prize Grand Challenge platforms allows customers to return, which provides an opportunity for Audrey to use information learned about them during their prior conversations (e.g., interests in particular movies or sports) to engage with them. However, returning customers also present a challenge to our generators, as customers may have heard some of the content before (e.g., retreading the previous conversation). Therefore, we tested whether the conversation score was higher for returning versus new customers.[6]

Figure 6 shows a plot of the mean and the spread of ratings for new and returning customers. New customers gave an average rating of 3.21, and returning customers gave an average rating of 3.40. While returning customers gave a higher average rating, the spread is much larger compared to new customers. The distribution of new and returning customers were about 85% and 15% respectively.

3.1.5 How do different topic modules and generators affect the engagement of the customer?

Audrey uses multiple modules for generating speech (§2.3.5) guided by conversation policy goals (§2.2.3 and §2.2.2). These generators each can have different effects on the engagement of the customer, based on their interests. Here, we measure engagement through the number of response made by a single generator—i.e., did the customer talk to this part of Audrey more. Figure 7a shows that customers consistently engaged more with the neural response generators than with other template or hybrid generators. The most-utilized generator, Topical Response, contains multiple topics, each could have different levels of engagement. Therefore, we examined the average number of turns for each topic, shown in Figure 7b. Among topics, season and fitness led to more conversation,

---

[6] Note that because customer IDs reset at the beginning of very month, we are unable to fully track all return visits and some first visits by a new customer may actually be returning.

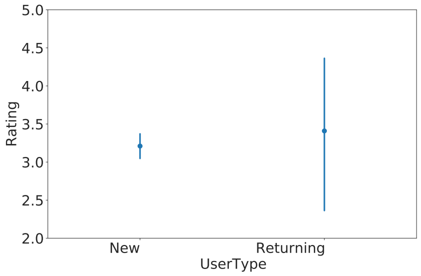
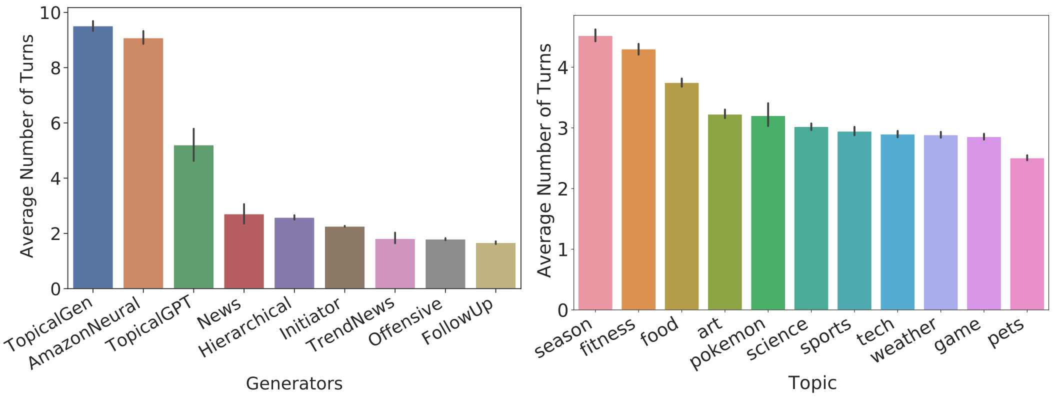


Figure 6: Point Plot based on new and returning customer comparison. Median ratings for returning customers were higher than the new customers.

with over a full turn more dialog. Customers engage with the majority of topics for at least three turns on average with the exception of pets which was slightly lower (2.50). We speculate that the pets topic results in fewer turns due to conversational redirection from people who don't have pets.



(a) Generators vs Average number of turns

(b) Topics vs Average number of turns

Figure 7: Overview of average number of turns vs generators and topics. a) The topical chats and neural generators engage customers the most. b) Among the topical chats, season and fitness topics get the longest conversations.

3.1.6 How do different topic modules and response intents affect the overall ratings?

Because Audrey's responses are generated by different initiators and topic modules, many factors can affect overall rating. In order to pinpoint the modules that are positively affecting our performance, we construct an OLS regression where we treat each conversation as an observation and include independent variables for (i) the number of turns from each generator, (ii) the particular conversation starter, (iii) generator expressions like asking for a customer to repeat themselves, (iv) Red questions uttered by the customer, and (v) the duration of the conversation in minutes. By using the customer rating as the dependent variable, this model let's us quantify the impact of each component.

The regression results, shown in Table 9, contain four main findings. First, the largest significant effect on the conversation come from longer conversations with the topical generators. Surprisingly, despite Pets having the shortest overall conversation (cf. Figure 7b) discussions on Pets has the largest positive effect on score among all topics. Other strong contributors were more narrow topics such as Art, Fitness, Food and Games. Second, our knowledge-rich hierarchical response generators (built on topic of EVI) did not significantly improve conversations, despite their ability go deeper into a topic based on domain knowledge. Third, we observe a statistically-significant negative effect for when Audrey uses Topical GPT directly to generate a response (not as a part of a strategy) based on what the customer has said. As this back-off strategy is evoked when Audrey cannot determine how to best proceed, this coefficient suggests that (i) a better strategy for handling non-sequitur comments by customers can improve conversation quality and that (ii) neural generators are not yet suitable to fully dynamically generate the conversation over long periods. Fourth, building upon the analysis of the initiators (§2.3.1), when controlling for all other factors, surprisingly, the way the conversation is initiated has no significant impact on the resulting score—though the most positive initiator, talking about weather, approaches significance with a large positive coefficient.

	β	Standard Error
Intercept	2.948***	0.104
Duration (minutes)	0.001***	0.000
Conversation Starter: None	0.095	0.105
Conversation Starter: Art	0.079	0.118
Conversation Starter: Fitness	0.045	0.110
Conversation Starter: Food	0.067	0.110
Conversation Starter: Games	0.071	0.117
Conversation Starter: Pets	0.071	0.117
Conversation Starter: Pokemon	0.132	0.144
Conversation Starter: Science	0.011	0.120
Conversation Starter: Season	0.184*	0.110
Conversation Starter: Sports	−0.073	0.123
Conversation Starter: Tech	0.043	0.120
Conversation Starter: Weather	0.170	0.117
Hierarchical Response: Books	0.022	0.017
Hierarchical Response: Movies	0.028*	0.016
Hierarchical Response: Music	0.013	0.022
Topical Response: Art	0.031**	0.013
Topical Response: Fitness	0.039***	0.007
Topical Response: Food	0.034***	0.009
Topical Response: Games	0.043***	0.014
Topical Response: Pets	0.055***	0.016
Topical Response: Science	0.027**	0.014
Topical Response: Season	0.028***	0.006
Topical Response: Sports	0.016	0.015
Topical Response: Tech	0.028**	0.014
Topical Response: Weather	0.018	0.014
Amazon Neural Generator	−0.006***	0.002
Follow Up Questions	−0.011	0.013
Goodbye Response	0.042*	0.024
Response to Offensive Comments	−0.048***	0.014
Repeat Response	0.047	0.032
Observations	16,404	
R^2	0.033	
Note:	$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$	

Table 9: OLS regression result for different intents that could effect the overall rating.

3.2 Experiments

Throughout the quarterfinals and semifinals periods, multiple A/B tests were done on the platform to test the effect of specific components. In the following tables, the average conversation durations are adjusted to exclude outliers. The collected data would have a couple data points magnitudes higher, as long as an entire day, which are believed to be a user leaving their device without ending the conversation.

3.2.1 Intimacy Experiment for transitions

Our question intimacy estimations modules enable us to potentially move from more causal to deeper questions during conversation. To test whether systematically increasing question intimacy leads to higher-ranked conversations, we randomly selected conversations to engage in one of two strategies:

(1) asking conversational questions in increasing order of intimacy or (2) asking the same questions in a random order, using the 76 questions from Section 2.3.2. This experiment was conducted over 4,104 conversations from March 28th to April 6th.

We hypothesized that by picking less intimate conversational questions for transitions in the beginning and gradually increasing the intimacy level of the question, customers would have a better feeling of the conversation that would translate into ratings. However, the results in Table 10 showed a statistically-significant drop in conversational score, when questions were asked in increasing order. We speculate that while the questions were intriguing, they were not always asked at times that lead to strong conversational coherence, which negated the effect of question ordering.

Variant	Avg. Feedback Rating	Feedback CI	Avg. Conversation Duration (seconds)	Avg. Conversation Duration CI
Random order	3.321	(3.287, 3.355)	223.22	(216.097, 230.343)
Increasing intimacy	3.201	(3.167, 3.235)	196.33	(190.076, 202.584)

Table 10: Mean conversation ratings and duration when transitioning between topics using questions ordered by increasing intimacy versus the same questions asked in a random order. Intimacy had no effect on customers' preferences for the conversation, but increasing intimacy led to a statistically-significant drop in conversation duration.

3.2.2 Trending News Conversation Generator

Our Trending News Conversation Generator (TNCG) allows us to inject interesting discussion points during the middle of a conversation when the Dialog Policy manager (§2.2) has identified that the customer's focus has drifted, potentially sparking more engagement. However, such transitions could seem out of place and jar the customer, especially with reliance on Topical GPT to generate subsequent conversation on recent news. Here, we performed an A/B experiment to test the effect of introducing recent news in a conversation by randomly varying whether the TNCG (§2.3.3 or Follow-up Generator (§2.3.2) was used to transition between topics. The experiment was run during April 21st to April 23rd across as sample size of 1348 conversations. Our results in Table 11 showed a statistically-significant increase in the average conversation duration with the TNCG. The generator had no significant effect on rating, the discrepancy which agrees with our findings in §3.1.2 that the two are weakly correlated. We speculate that TNCG generally increased conversation duration based on the interest of recent news over templated topic changers, but that its under-preparedness to discuss COVID-19 caused inconsistency with the final ratings. We observed a general lack of interest in discussing coronavirus news, which encompasses nearly our entire pool of content, despite selecting for positive-only news.

Variant	Avg. Feedback Rating	Feedback CI	Avg. Conversation Duration (seconds)	Avg. Conversation Duration CI
Trending News Gen.	3.276	(3.217, 3.330)	224.14	(212.069, 236.211)
Followup Generator	3.251	(3.194, 3.308)	151.00	(109.665, 132.3348)

Table 11: Experiment on trending news generator and follow up generator.

3.2.3 Experiment on Providing Context to the Neural Generators

The neural response generator (NRG; §2.3.9) conditions its output based on prior context. More context can potentially provide richer information to craft a conversational arc matching the current trajectory; however, selecting prior turns that cover multiple topics may muddy the output. Therefore, we conducted an A/B experiment to test the effect of different types of context provided to the NRG. The first condition uses all prior turns on the current topic as context, while the second always selects the 5 previous turns as context.

The experiment ran March 20th to March 28th, over 3,538 conversations, with results shown in Table 12. Including only on-topic context for the NRG attained a mean rating of 3.308 compared with the 3.196 rating for using a fixed-size context. We observed the only on-topic context to be statistically different from fixed sized content, both in terms of higher average feedback ratings and lower average conversation duration. The inverted trends between ratings and conversation duration is unexpected, but this again corroborates with the findings in §3.1.2 that suggest only a weak correlation between the two. Nevertheless, we view this as a useful guide for the importance of conditioning on a more topically-narrow context to improve response quality and coherence.

Variant	Avg. Feedback Rating	Feedback CI	Avg. Conversation Duration (seconds)	Avg. Conversation Duration CI
Only On-topic Context	3.308	(3.287, 3.329)	208.06	(204.130, 211.990)
Fixed-size Context	3.196	(3.181, 3.211)	242.73	(240.122, 245.338)

Table 12: Ratings Change on Introducing Dynamic Context Length for GPT-2

4 Conclusion

We designed an open-domain social conversation system, Audrey, that achieved a cumulative average rating of 3.25 out of 5 in the the semi-finals. Audrey was designed with the goals of engaging with customers on a personal level by adapting to their preferences, interests, and personality. To achieve these goals, we developed a large collection of NLP modules for language understanding (§2.1, dialog management (§2.2), and response generation (§2.3) that create a diverse conversational landscape intended to evoke delight. Throughout the competition Audrey engaged with tens of thousands of customers and during semifinals achieved mean rating of 3.25.

5 Future Work

Given time constraints from late deployment in semifinals, we were not able to perform rigorous experimental analysis on the innovative features released recently like Personal Understanding Module (PUM) and Reinforcement Learning (RL) based Topic Selection and some new topical modules. We can conduct more analysis if we remain in the competition after semi-finals and there are certain areas that we would want to improve our system on. We will aim to improve the Personal Understanding Module (PUM) so that it can better suggest topics based on customers' likes and dislikes and create an adaptive and unique conversation experience for them. Lastly, we also plan to improve our reinforcement learning model to build a better dialog policy for topic selection and conversation content planning.

6 Acknowledgement

We would like to acknowledge the help from Amazon Alexa Prize Challenge team and all Amazon Engineers for their constant technical and financial support and cooperation during this challenge.

References

[1] Arthur Aron, Elaine N Aron, and Danny Smollan. Inclusion of other in the self scale and the structure of interpersonal closeness. *Journal of personality and social psychology*, 63(4):596, 1992.

[2] Petter Bae Brandtzaeg and Asbjørn Følstad. Why people use chatbots. In *International Conference on Internet Science*, pages 377–392. Springer, 2017.

[3] Justine Coupland. *Small talk*. Routledge, 2014.

[4] Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*, 2011.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Hao Fang, Hao Cheng, Maarten Sap, Elizabeth Clark, Ari Holtzman, Yejin Choi, Noah A. Smith, and Mari Ostendorf. Sounding board: A user-centric and content-driven social chatbot. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 96–100, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

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

[8] Devamanyu Hazarika, Soujanya Poria, Roger Zimmermann, and Rada Mihalcea. Emotion recognition in conversations with transfer learning from generative conversation modeling. *arXiv preprint arXiv:1910.04980*, 2019.

[9] Matthew Honnibal and Mark Johnson. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[10] C.J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. 01 2015.

[11] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[12] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June 1993.

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

[14] United States. Bureau of the Census, United States. Bureau of Labor Statistics, and National Endowment for the Arts. Survey of public participation in the arts (sppa), 2012 [united states], 2015.

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

[16] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. Towards empathetic open-domain conversation models: a new benchmark and dataset. In *ACL*, 2019.

[17] Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2017.

[18] Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. Transfertransfo: A transfer learning approach for neural network based conversational agents, 2019.