# Bernard: A Stateful Neural Open-domain Socialbot

**Bodhisattwa Prasad Majumder**
CSE, UC San Diego
bmajumde@eng.ucsd.edu

**Shuyang Li**
CSE, UC San Diego
shl008@eng.ucsd.edu

**Jianmo Ni**
CSE, UC San Diego
jin018@eng.ucsd.edu

**Huanru Henry Mao**
CSE, UC San Diego
hhmao@eng.ucsd.edu

**Sophia Sun**
CSE, UC San Diego
shs066@eng.ucsd.edu

**Julian McAuley**
CSE, UC San Diego
jmcauley@eng.ucsd.edu

## Abstract

We propose *Bernard*: a framework for an engaging open-domain socialbot. While the task of open-domain dialog generation remains a difficult one, we explore various strategies to generate coherent dialog given an arbitrary dialog history. We incorporate a stateful autonomous dialog manager using non-deterministic finite automata to control multi-turn conversations. We show that powerful pretrained language models are capable of generating coherent and topical responses in the presence of grounding facts. Finally, we implement Acknowledge-Retrieve-Reply strategy to combine template-based and neural dialog generation for greater diversity and increased naturalness. Extensive human evaluation shows that the combination of generative models and retrieval models in a stateful dialog machine can achieve desired user experiences in terms of topic diversity and engagingness, as showed in extensive human evaluation.

## 1 Introduction

From the original Mechanical Turk to today's human-like chatbots [1], humans have been fascinated with the idea of interacting with machines in human modalities rather than via programming. One of the first chat programs, ELIZA [25], elicited great fanfare and inspired the broader anthropomorphism of computers. While the broader line of work in chat systems has focused on text-based systems, recent dialog systems and AI assistants operate mostly on speech. From 2016 onwards, major industry research labs and corporations (Amazon, Google, Facebook, and Microsoft to name a few) are heavily investing in such user-friendly chat systems and assistants.

Building such large-scale dialog systems requires extensive training and evaluation capabilities which are often lacking in academic and smaller industrial settings. Advances in the field of AI both inspire hope and also raise expectations (sometimes excessively) for nuanced, personalized, and omnipresent conversational systems.

We propose a framework—Bernard—for building such dialog systems with a focus on fluidity, diversity, and knowledge-rich conversation. By incorporating non-deterministic finite automata (NFA), Bernard supports an autonomous dialog manager which seamlessly guides users through conversational states. The framework is modular and easily extensible for developers with a broad idea of conversational flow and logic. Bernard additionally incorporates several powerful neural generative dialog models capable of generating coherent and diverse responses based on a dialog history and factual knowledge.

We present a socialbot built with this framework—also named Bernard—as part of the 2019-2020 Amazon Alexa Prize Socialbot Grand Challenge 3. In this technical report, we describe the components of our dialog system, its motivations, and observations on dialog structure from our interactions

3rd Proceedings of Alexa Prize (Alexa Prize 2019).

with a significant number of real-world socialbot users during the span of the Challenge. We successfully implement an Acknowledge-Retrieve-Reply strategy to guide engaging response generation with meaningful information.

**Goal-Oriented vs. Open Domain** Recent efforts and advances in chatbot technology are mainly geared towards goal-oriented tasks such as restaurant reservation or flight booking [5]. These domain-limited systems have the advantage of a clear optimization criterion (i.e. success in booking), but do not generalize to spontaneous conversation. Open-ended dialogs [23] generally deal with multiple domains and various subtopics, with no clear goal present for either participant of a conversation. To address this issue, frameworks like Bernard optimize broadly for conversation quality metrics including naturalness and engagement. These can take the form of statistical likelihood of dialog and conversation duration. The Amazon Alexa Prize is one example of an industry-led effort to advance in this field of open-ended conversational chatbots.

**Chatbots: The Production Setting** Building each component of an open-domain socialbot takes significant effort including data collection, modeling, and the evaluation and feedback pipeline. This is to speak nothing of full end-to-end approach, which we found to be untenable, leading to Bernard adopting a more modular NFA-based framework for guiding conversation flow. Automatic evaluation of dialog systems is extremely difficult, with human evaluation becoming a standard. The Alexa Prize Challenge provides a unique opportunity to test our methods and strategies at scale. Our primary goal with Bernard was to make an engaging and personalized chatbot, and while we have implemented several techniques to advance in these directions, we have but scratched the surface during the limited time of the Challenge. The demands of a production service (load-bearing, legal limitations, and a hugely diverse audience) demand a great deal of engineering work overshadowing at times more exploratory research. Nonetheless, we believe the contributions of Bernard will remain relevant and provide promise of better personalized and engaging open-domain socialbots.

## 2 System Design and Architecture

The Bernard open-domain chatbot is built on the Amazon Conversational Bot Toolkit (`cobot`) [13] —a framework that connects the bot code with AWS Lambda[1] to serve the chatbot to users seamlessly with automatic compute scaling for component modules.

We rely heavily on cobot's state manager, which stores per-session and per-turn information about user utterances and Bernard's internal state to DynamoDB[2].

In this section we cover the individual components of the Bernard system.

### 2.1 Persistent Storage

We primarily leverage two forms of persistent storage relating to Bernard. First, we utilize Amazon DynamoDB, a NoSQL database, for caching our bot's internal state, as well as a variety of topical conditioning information for our dialog models. This includes Reddit[3] threads, news snippets from National Public Radio (NPR)[4] and Washington Post (WaPo)[5], and structured information about movies from the Internet Movie DataBase (IMDB)[6].

Additionally, we store static training data and model checkpoints via Amazon S3[7].

### 2.2 Automatic Speech Recognition (ASR)

We leverage the built-in ASR service from Amazon Alexa to transcribe user utterances. We find that for short user utterances, Alexa ASR performs well, with observed errors from our informal tests

[1] `https://aws.amazon.com/lambda/`
[2] `https://aws.amazon.com/dynamodb/`
[3] `https://www.reddit.com/`
[4] `https://www.npr.org/`
[5] `https://www.washingtonpost.com/`
[6] `https://www.imdb.com/`
[7] `https://aws.amazon.com/s3/`

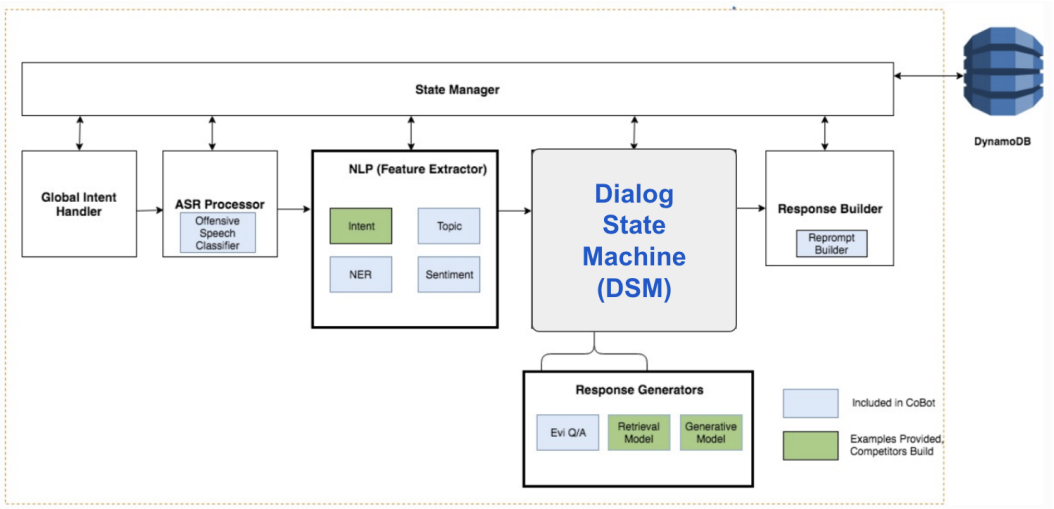



Figure 1: Bernard—a stateful neural socialbot framework. Figure adapted from [13].

primarily due to pluralization or singularization. We believe there may be ASR performance concerns for more complex conversational interactions based on observations of other ASR systems in the market, but are not able to measure this directly at the current time. In future work, we will consider the problem of noisy transcription and error-correction in ASR, leveraging the expanded Amazon Topical Chat[9] dataset incorporating a set of mislabeled ASR candidates.

### 2.3 Natural Language Understanding (NLU)

Central to Bernard's function is the ability to parse and understand user utterances, in order to retrieve and generate appropriate responses.

One key insight we observed throughout the competition was regarding the brittleness of a modular approach for NLU: while it is extremely difficult to train end-to-end models to return the wide variety of necessary NLU outputs, off-the-shelf, pre-trained models for NLU often adapt poorly to open-domain conversation, when those models were not trained for this particular setting.

We have observed that off-the-shelf methods for sentiment and intent classification work well to support more structured and rigid response generation. However, we have in particular elected to develop our own topic extraction module based on the dialog management framework using a Nondeterministic Finite Automaton (NFA) in conjunction with the default cobot Named-Entity Recognition (NER) and Amazon Alexa Skills Kit (ASK) intent classifier and dialog managers[8].

#### 2.3.1 Sentiment Classification

In Bernard, we use the Valence Aware Dictionary and sEntiment Reasoner (VADER) sentiment classifier[11], incorporating a social-media-specific sentiment lexicon. The VADER classifier is attuned specifically for microblog-style settings, but we have found empirically that it is appropriate for general sentiment classification in the open-domain chat setting as well. Specifically, we use sentiment scores of user utterances to inform our acknowledgements of user input (see Dialog Strategies section).

#### 2.3.2 Utterance Encoding

Many of our NLU modules rely on a contextual encoding of user utterances (e.g. topic classification). We utilize two pre-trained methods for utterance encodings: a weighted composition of GloVe embeddings of constituent words [2, 18] as well as SentenceBERT whole-utterance encodings [20]. While SentenceBERT encodings showed slightly better overall performance for NLU tasks, we

[8] `https://developer.amazon.com/en-US/alexa/alexa-skills-kit`

elected to use weighted GloVe and FastText [4] embeddings to decrease latency while maintaining an acceptable level of performance.

We note that historical user and bot utterances used by our neural generative models are encoded by the model itself, and the utterance encodings described in this section are used exclusively in the NLU pipeline.

#### 2.3.3 Dialog Intent Classification

Dialog intent classification is an essential step in NLU that helps to elicit different dialog strategies. We leverage the pre-trained model provided by the Alexa team [12]. Bernard also implements additional classifiers to deal with custom intents in our ontology that are not covered by the pre-trained model, such as personality-related and domain-specific content generators. To implement these classifiers, we start with rule-based regex matching and further extend them with the new user utterances collected during the competition.

#### 2.3.4 Topic Extraction

When crafting a bot response, of particular importance is determining the topic of conversation at hand. By narrowing the domain, we are able to retrieve relevant world facts and news for a conversation, as well as direct users to more appropriate sections of our dialog state machine. Our topic extraction pipeline consists of two parts: 1) building a robust topic ontology for open-domain conversation, and 2) a clustering and K-Nearest-Neighbor (KNN) based algorithm for topic classification.

**Topic Ontology** We built a large corpus of topics by extracting entities from 78K+ Reddit thread titles, combined with the curated 300-entity list from Topical Chat. For each entity, our dataset contains multiple (up to several hundred) related facts. We then clustered topics by computing the centroids in our topic table using FastText [3] embeddings, averaging the words in each phrase to obtain a phrase embedding.

**Topic Classification** During a conversation, our topic module must be able to classify the current conversation's topic. We classified topics by first extracting noun phrases from the last turn of the user. Then, we used FastText [3] to again extract phrase embeddings for each noun phrase. We use an optimized nearest-neighbor search to compare noun phrases from the current conversation with our database of topics to extract the most relevant five. Our grounding information is then sampled from all facts linked to one of these neighboring topics, weighted by similarity score.

### 2.4 Dialog Manager

Dialog management is one of the key components of long open-ended dialog. It is also different from goal-oriented dialog where there is a fixed goal to achieve at the end of the conversation. For open-domain dialog, the aim is to have a diverse, engaging and longer conversation. To facilitate that, we introduce a 'Dialog State Machine' (DSM) - a finite state machine in the form of a non-deterministic finite automaton (NFA) that can control the high-level flow of the conversation. After a high-level flow is determined given a dialog history, we perform low-level dialog management to navigate the user.

#### 2.4.1 Dialog State Machine

We define a non-deterministic finite automaton with the following -

- **A finite set of states**: Introduction, Small Talk, Clarification, Topical Chat and a number of Domain specific states
- **A transition function**: a function that uses dialog history to determine a probability of transition from one state to the other.
- **An initial or start state** which is always Introduction.
- **A final state**, Stop.

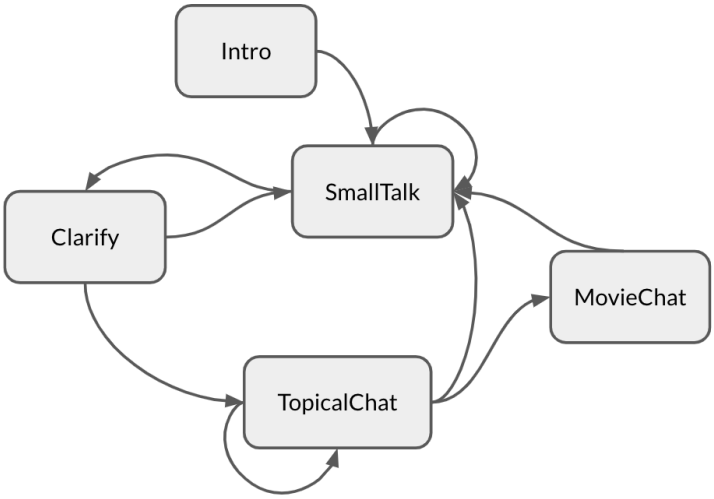



Figure 2: Dialog State Machine as a non-deterministic finite automaton

### 2.4.2 States

We observe a general high-level flow and structure in human-to-human and human-to-bot conversation. A typical conversation starts with introductions and atopical, phatic small talk, occasionally referencing contextual events. This transitions into a deeper, single-conversation topic. The speaker may also ask clarification questions to support their understanding of the conversation and their counterpart. Following this pattern, we defined major top-level states: Introduction (Phatic), Small Talk, Clarification, Topical Chat; as well as domain-specific states. Each state is responsible for carrying out a specific type of conversation and seamlessly transitioning to other states when appropriate.

**Introduction** An introduction state is invoked when a user first expresses interest in conversation. Generally, the bot introduces itself with a fixed introduction message and asks the name of the user. We do not store the name of the user, electing instead to respond that it is a good name. We choose from a set of fixed responses for this acknowledgement.

We follow by asking a question about recent life events—we observe that users are most comfortable speaking about their own experiences when freed of any topical constraints. We sample a time frame from the last month to the last day, and ask about any novel happenings in this period. The introduction state always moves into Small Talk (phatic conversation).

**Small Talk** This state is responsible for general phatic communication—literally, small talk: a part of the conversation when both speakers are exploring a suitable topic to converse with. We noticed discomfort in our beta testers when mimicking dialog patterns of socialbots from the previous iteration of the Alexa Prize challenge, which tend to initially steer the user into conversations about video games, movies, or jokes. Bernard allows the user more agency—rather than immediately jumping into extended single-topic conversation, we organically seed the conversation with Bernard's 'experiences' and ask the user about their life events.

The user may return to this Small Talk state later in the conversation when they indicate boredom or the conversation becomes atopical. Our DSM design allows us to invoke Small Talk from any domain-specific or topical state, controlling and adding flexibility to the flow of dialog.

In Small Talk, we apply our Acknowledge-Retrieve-Reply framework to gently warm-start the conversation. We use a neural dialog model, as described in Section 2.7.6, to generate an acknowledgement. We then utilize a truncated dialog history for small talk along with retrieved facts to generate a content-rich response, as seen in the underlined section of Table 1. We incorporate 'life experiences' of Bernard, curated by manual annotators.

Small Talk at its core comprises an NFA between nine topical states: Food, Hobby, Place, Fashion, Movie, music, Society, Book, and Introduction. The Introduction state indicates entering phatic

| | |
|---|---|
| **Bot** | You have a nice name. What's your highlight from 2019? |
| **User** | Thanks. I travelled a lot in 2019. |
| **Bot** | That's great to hear. **I traveled to 10 countries in 2019. Out of all of them, I really liked Tonga and its beautiful beaches.** What's your favorite island? |

Table 1: Sample conversation in Small Talk. The underlined part of the final response is acknowledgement to what user said. The **blue-bold** indicates a snippet of Bernard's personal experience, predefined and rest of the blue part is a follow-up question to user to talk on that topic. To respect users' privacy, this is not a real user conversation.

conversation from another top-level state such as Topical Chat or Clarification. We have manually crafted a list of multiple conversation snippets for each transition between topical states within Small Talk, totalling 153 dialogs referencing experiences and expressing opinions. We include a sample of these experiences and opinions in Table 2.

| **Topical Flow** | **Bernard's experience** |
|---|---|
| Intro → Food | Last week I ordered delivery from this Italian place. We thought it was going to be generic, but they had some crazy good pasta! What're your favorite kinds of noodles? |
| Food → Book | I think staying at home makes me hungrier for some reason. So much that I'm trying to find food books to read and distract myself. Do you know of any good ones? |
| Book → Music | Sometimes I wonder how good authors would be at writing song lyrics. Do you think they would write meaningful lyrics or if they would sound good at all? |
| Music → Movie | I like all sorts of music, but I'm still not quite sold on Mongolian throat-singing. Those musicians make sick music videos though. Have you watched anything good recently? |
| Movie → Hobby | One of my favorite cult films is Space Jam. It never gets old for me! I wonder if it would have been as successful if Michael Jordan played another sport. What do you think separates a sport from a hobby, if anything? |
| Hobby → Society | Sometimes I feel really tired and don't want to use my brain. But I've always got it in me to curl up with a newspaper and learn about what's happening in the world. What do you think about the state of the world right now? |

Table 2: Sample curated 'experiences' for Bernard. The content reflects recent events, and interesting facts, inviting deeper conversation.

After the Introduction, Bernard engages in Small Talk for 3-4 turns. This allows the user to familiarize themselves with the bot before starting a deeper conversation.

While in Small Talk, we constantly look for a topic, as described in Section 2.3.4. When the topic extraction confidence passes a certain threshold, we break out of Small Talk and transition into other states. From Small Talk, the conversation can go to Clarification question, Topical Chat and other domain specific Dialog Managers depending on the user query.

**Clarification** We note that all of the approaches we have tried for topic discovery work well in a test setting, but achieve various levels of robustness to the topical diversity of real conversation. To account for this issue with our unsupervised topic extraction models, we confirm our candidate topics with the user in the Clarification state. Here, we ask a clarifying question about which of the detected topics the user truly wishes to discuss. We generate these questions from templates, such as: `It seems like you are interested in <topic>. Did I understand you correctly?`

An affirmative user response will transition the conversation to Topical Chat or a domain-specific state, while a negative or indecisive one indicates a return to Small Talk is likely necessary.

**Topical Chat** Once Bernard is confident about a topic based on the topic extraction model confidence or an affirmative user response, we move to Topical Chat. In Topical Chat, our bot is capable of talking about a specific topic or set of related subtopics. We use templated, fact-based (Section 2.6), and neural generation based models (Section 2.7) to create natural and diverse user responses.

Much like the Small Talk module, the Topical Chat module searches for new candidate topics in each turn of dialog via the topic extraction module. When the topic extraction module asserts low confidence, we transition back to Small Talk to diversify the conversation.

**Domain-specific states** Domain specific states are responsible for conversing on specific domains such as movies or news. These states themselves consist of sub-NFAs. We additionally maintain a state for Bernard's Favorites to specifically answer questions like 'Who/what is your favorite ___?'. We describe these states in detail in Section 2.5.

#### 2.4.3 State Transitions

Figure 2 depicts the possible transitions for the major states in our dialog state manager. While the set of possible next states is fixed for each NFA state, the transition probabilities between each pair is not necessarily set. We experimented with transition policies for the Small Talk, Movie Chat and Topical Chat states, and note our observations below. Across all policies, there were ways for the user to explicitly note a change in state (e.g. 'Stop talking about movies' to transition from Movie Chat to Small Talk).

Our initial policy presented a uniform probability across all state transitions, in order to add diversity to the conversation duration within each module. Under this policy, we received generally positive feedback about dialog diversity but conversations segments tended to remain in Small Talk for a very short period of time. As we received the most positive feedback about Small Talk, we experimented with setting a minimum duration for the module in question. During this duration, Bernard accumulates topics touched upon by the user but does not directly transition into the Clarify state.

We also tracked the states for each user conversation, and assessed the global likelihood of transitioning from Topical Chat to Small Talk. We focused on this transition as it was the most common transition seen across conversations—dialogs with Bernard tended to move between Small Talk, Clarify, and Topical Chat as users' interests and attention changed. We experimented with using this global likelihood as a prior for our transition probability between Topical Chat and Small Talk, but found no noticeable difference. We hope to explore this direction in more detail in future research, as this experiment necessarily co-occurred with a consistent stream of upgrades to bot functionality that could have acted as confounding factors.

We note several advantages to our NFA-based approach to conversation structure:

- The DSM is built for long, fluid conversations, allowing us to cycle between states with learned and dynamic transition probabilities.
- The DSM can accommodate arbitrary conversation templates by altering state transition probabilities and adding new states independent of other dialog modules.
- The DSM ensures easy transitions from one state to another other, encouraging users to have open-ended chitchat.
- The DSM encourages topical diversity, and supports warm-starting and cooling-off during the transition from one topic to the other.

### 2.5 Domain-specific Dialog Managers

In addition to the top-level dialog manager, Bernard additionally incorporates domain specific dialog managers which operate on the nuances of the individual domain. We observed from our user feedback that users actively look for discussions on specific domains such as movies. Since these conversation topics can rely heavily on domain-specific ontologies and may require specific handling of domain specific entities and their relationships, we built domain specific dialog managers (NFAs themselves).

#### 2.5.1 Movie Chat

Movie Chat is solely responsible for conversing about movies. We collected a movie knowledge graph from the Internet Movie DataBase (IMDB) API[9], containing ratings, movie names, cast, crew,

[9] `https://www.imdb.com/interfaces/`

and synopses. We support in-depth conversations about movies with our Movie Chat NFA, comprised of three states:

**Movie Info** We retrieve information regarding a movie, its cast, rating and genre, and generate a templated response conditioned on these facts. For example, we refer to a movie with an IMDB rating over 8 as a 'masterpiece' and ask the user for their favorite parts. This state is triggered when a user mentions a specific movie title (obtained via regular expressions and fuzzy matching).

**Ask Genre** If a user has not mentioned a specific movie or indicates they are bored of this movie, we ask them instead for their favorite genre of film. We use the confidence of our topic extractor and semantic similarity of the user utterance with previous topics as a proxy for topical interest. When asking for genres, we provide a few starting options, always including at least one genre that has not been mentioned in the course of the conversation.

**Recommendation** Once a genre is identified from the user's response, we provide a recommendation—either explicitly, or by asking a question about one of the more popular movies in the genre. If a user consents or acknowledges our response, we transition back to the Movie Info state.

Such a cycle continues until the user wants to switch to another topic (global dialog manager state transition) or if we stay for more than 3 cycles in Move Chat.

### 2.5.2 Favorites

Users are often interested in Bernard's opinions—specifically, the bot's favorite items from certain categories. They frequently ask Bernard things like *What is your favorite Movie?*. To answer these questions and build a consistent persona for Bernard, we manually curated a bank of 'favorite'-style preferences, with a selection shown in Table 3. We use regular expressions to detect a question that is asking for this type of response, and match the topic to the table to retrieve an appropriate answer. We often 'flip' the question back to the user, asking them for their favorite in the same category.

| Topic | Bernard's favorite |
|---|---|
| Movie | Shrek, a heartwarming bildungsroman featuring the growth and maturing of the world's most lovable Mike Meyers alter-ego. |
| Video Game | Super Mario 64 for sure. I've always wanted to be an Italian plumber but alas I wasn't born Italian nor did I ever find a plumber willing to take me under their wing. Mario lets me live that fantasy every day. |
| Food | One slice of soda bread with fried spam and arugula. Serve it up with a drink and you've got yourself one heck of a light but delicious lunch. |

Table 3: Sample snippets of Bernard's favorites, written by human annotators.

## 2.6 Fact Retrieval

One of the strengths of a chatbot like Bernard is its access to large stores of factual information. While previous entries in the Alexa Prize challenge have treated fact retrieval as a task of surfacing facts explicitly to the user in order to continue a conversation [14], we rely on fact retrieval to condition our model, and create responses with a more subtle factual grounding.

Our fact retrieval model is tightly coupled with the topical classification module (Section 2.3.4)—each topic cluster contains a selection of facts from Reddit, news sites, and the Topical Chat dataset corresponding to topical phrases and subjects within the cluster. We retrieve up to five facts per user utterance, conditioned on the conversation history with emphasis on the most recent utterance, and pass these as input to our knowledge-grounded neural dialog generator.

The one exception to this implicit usage of facts is in the domain of news. When asked a question in the form of *What happened in China today?*, Bernard will respond in a way that contains the verbatim headline of a piece of news. In future work, we wish to explore how to incorporate paraphrasing models to expand the diversity of returned facts and seed generation therewith.

### 2.7 Generation Models

We use a combination of hand-crafted response templates, Amazon-provided factual and limited-domain generation tools (EVI), and neural generative models to create a response for Bernard. This response is then passed to Alexa's text-to-speech (TTS) module, turning into the bot response heard by the Alexa user. Our dialog generation modules are based on evolving research in unconditional, knowledge-grounded[7, 8], and speaker-conditioned [17, 27] dialog modeling.

#### 2.7.1 Hand-crafted Templates

We have seen a great deal of success initiating conversations with a set of modular hand-crafted responses that fit into rough templates, particularly in our Smalltalk module, as seen in Table 2.

These response templates are aligned in the Acknowledge-Retrieve-Respond framework (described in the Dialog Strategy section), and we use these responses to add color to a conversation as well as prime the user with a sense of Bernard's persona. These often give hints as to our bot's preferences, and serve to seed a set of topics for the remainder of the conversation.

We observe that users are eager to respond to Bernard after hearing these responses, and that it improves their reaction and willingness to engage in longer conversation with a subsequent neural or retrieval-based response generator. In particular, we have received feedback from a variety of beta testers that crafted responses bearing surprising characterizations (e.g. the movie Shrek as a *heartwarming bildungsroman*), irony, and reminiscence provide a great deal of enjoyment as well as a higher tolerance for unusual statements that may be generated from our neural models down the line.

These qualitative measures and patterns for dialog quality are the subject of open research [22] in the field of dialog modeling, although for our purposes a modest corpus of hand-crafted responses and bot experiences provides sufficient diversity for our smalltalk module.

#### 2.7.2 EVI

We used EVI as a strong fallback response generator to handle latency- and load-related outages for our dialog model services. EVI, a service provided by Amazon Alexa, is a general question answering service. It is able to handle factual queries like *How deep is the Pacific Ocean?* as well as a limited set of personal queries such as *What is your favorite color?*

While it is designed to provide appropriate responses for some common phatic conversation starters (e.g. *How's it going?*), EVI is a non-contextual response generator that cannot vary its generation based on user or conversation history. This limits its usefulness as a full-fledged response generation mechanism, and its brittle nature often leaves it response-less when it encounters more complex utterances (e.g. *Okay well then tell me the capital of New Zealand*).

As a result, we transitioned into using neural dialog models for improved expressivity, personalization, and incorporation of knowledge into conversation.

#### 2.7.3 Transformer Language Models

Our neural models are based on the transformer architecture [24]. We train our models with a conditional language modeling objective to maximize the probability of target sequence $T = x_1, ..., x_m$ given a source sequence $S = w_1, ..., w_n$

$$P(T|S) = \prod_{i \in \{1...m\}}^{N} P(x_i|w_1, ..., w_n, x_1, ..., x_{i-1})$$

We experimented with three different neural models for generation—two using a decoder-only architecture in the vein of GPT2[19] which has been shown to be capable of high-quality open-domain text generation, and one based on an encoder-decoder framework[15] to explicitly encode dialog history and grounding documents.

#### 2.7.4 Reddit-trained Transformer Decoder

For the initial phase of the competition, we utilized a pre-trained checkpoint of DialoGPT[28], a GPT2 model trained on a set of 147M Reddit threads covering 1.8 billion words in total. While this model is not explicitly trained on dialog, we find that Reddit threads are a sufficient proxy for dialog for purposes of brief acknowledgement phrases, and this was the capacity in which we used DialoGPT for much of the pre-quarterfinals period.

We later fine-tuned DialoGPT on Topical Chat and the Interview corpus[17] to allow for longer acknowledgements and as a neural fallback generator for phatic turns of conversation. All variants of DialoGPT that we used were conditioned purely on dialog histories and contained no external grounding information.

#### 2.7.5 Speaker-conditioned Transformer Decoder

While GPT2 variants were able to generate coherent responses, the generation quality often tended toward generic phrases. To create a more fluid and welcoming user experience, Bernard moved towards using models that allowed for personalized dialog generation.

Here, we adapted the speaker role-conditioned conditional generative model from [17]. We represent users and guests with a separate speaker role embedding, with the language modeling objective maximizing the likelihood of a user utterance $U_{t,b}$ given previous utterances by the guest $U_{1...t,a}$ and bot $U_{1...t-1,b}$:

$$P(U_{t,b}|U_{1...t,a}, U_{1...t-1,b}, b)$$

We evaluated the generation quality and appropriateness across many conversations with beta testers, and the encouraging feedback led us to transition from a 50:50 split of using EVI and neural models for acknowledgement generation to fully using the speaker-conditioned neural dialog model.

#### 2.7.6 Denoising Transformer Encoder-Decoder

To improve our neural generated responses, we trained a Transformer encoder-decoder model on the Topical Chat Dataset[9] to enable us to provide knowledge-grounded conversational responses. These neurally generated responses enabled us to provide more diverse bot responses when compared to templated or deterministic generation (e.g. EVI). Due to the small size of the Topical Chat dataset, we fine-tuned the BART [15] pre-trained model on this dataset. BART is a Transformer encoder-decoder that is trained on the denoising task, where some inputs of the encoder are masked and the decoder must reconstruct these masked inputs.

To train the model, we formatted the knowledge grounding from the Topical Chat dataset as the encoder input, and used the full conversation as the decoder input. The decoder is trained to predict the next token given the conversation. The design of the input format in this manner has several practical advantages: 1) Grounded knowledge only changes between topic transitions—meaning that we can cache the encoder hidden states once they are computed, and it can be persisted through several turns of conversation; 2) Training the conversation in this manner is more efficient, as we can train on all turns of a full conversation in a single gradient update iteration.

We trained our model for 100 epochs and our model reached a final validation perplexity of 14.8 on the Topical Chat validation set.

## 3 Evaluation

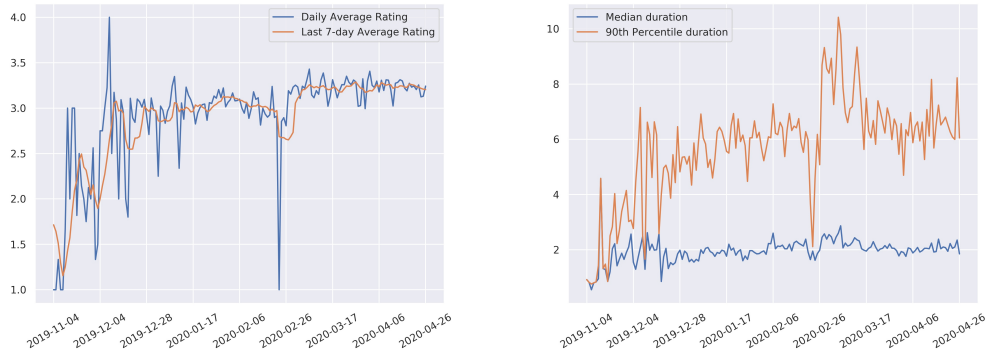
We used the Amazon Cloudwatch[10] infrastructure logging tool to track performance, as well as spot-check conversation quality.

Figure 3a shows the trend of Bernard's daily and last 7-day average ratings during the competition. The chatbot's performance improve gradually and reached 3.25 on average. We observe that due to downtime from load-related issues and content violations, the graph displays a significant lagging drop in L7D performance. To minimize the impact of these easily avoidable issues, we focused

[10] `https://aws.amazon.com/cloudwatch/`

on improving our abusive/blacklist content classifier. Figure 3b shows the trend of Bernard's daily conversation duration. Our chatbot's median conversation duration is about 2 minutes and the 90% percentile of conversation duration is about 6 minutes.

Key milestones include the introduction of our Small Talk NFA which led to an 8% improvement over a baseline prior rating of 3.0, as well as the complete DSM, which led to a spike in quarterfinals ratings, stability in the semifinals around 3.25, and a 38% increase in median conversation duration.




(a) (b)

Figure 3: (a) Daily average rating and last 7-day average rating; (b) Median duration and 90-th Percentile duration of daily conversations in minutes.

Table 4 contains summary statistics of conversations from April 22 to April 29, 2020. On average, the number of turns achieve 9.70, among which the average number of words per turn from Bernard and users are 27.42 and 3.07, respectively. We observe that while it remains challenging to elicit long utterances from the user, our framework is able to speak at length, and engage users in multi-turn conversations.

| | Mean | 50th Percentile | 90th Percentile |
|---|---|---|---|
| # of turns | 9.70 | 7.0 | 20.0 |
| # of words/turn from Bernard | 27.42 | 27.50 | 34.25 |
| # of words/turn from users | 3.07 | 2.80 | 5.00 |

Table 4: Statistics of the conversations from the last 7 days.

### 3.1 Beta Testing

Since we could not access the immediate feedback from Alexa users, we created a pool of beta testers unaffiliated with the project to receive concrete and timely human evaluation. For each beta participant, we made sure that the only information given to them was that Team Bernard was working on an open-domain chatbot served via Amazon Alexa, and the invocation command for our ASK skill.

To seed the pool of testers, we reached out to friends, friends of friends, and family members thereof in addition to other university students to ensure that our beta testers cover a wide variety of backgrounds and levels of familiarity with technology / chatbots.

Due to this scheme, beta testers were somewhat familiar with team members' senses of humor and speaking styles. While this ran the risk of us changing our bot to overfit the beta testing pool, this also helped us adversarially tweak templates to accommodate a more generally appropriate personality for Bernard.

Nonetheless, there remained a significant disparity between beta tester reactions and broader Alexa customer ratings and conversation traces, leading us to mainly use beta testing as a method of sanity checking smaller changes and evaluating larger module additions.

### 3.2 Feedback Loop

We bucketed conversations in three general categories: 1.0-rated conversations, 5.0-rated conversations, and those in between. Our working strategy was to manually review conversations each day, with a 60-20-20 split sampled from each bucket, respectively. Each member of the team independently reviewed each conversation, and we combined notes for general conversational patterns and issues with Bernard.

This feedback loop was particularly helpful in the Quarterfinal stage, where we were able to secure a place in Semifinals after significant overhauls to the DSM and Small Talk modules improved with feedback from beta testers and daily user conversation traces.

## 4 Using Alexa Prize Data

We primarily used the Alexa Topical Chat dataset [9] for model training, topic identification, and fact retrieval. In this vein, we made use of the dialog corpus, fact corpus, as well as the aligned whole corpus to train a knowledge-grounded dialog model.

**Dialog Corpus** By looking at the dialogs themselves without grounding or topic information, we treat Topical Chat in a manner similar to DailyDialog [16] and other proxies for spontaneous human conversation. Accordingly, we use this section of the data to fine-tune large pre-trained dialog models, with the expectation that world knowledge has been stored implicitly in the model parameters.

**Topic Corpus** We predicated our topic model on the key entities within the Topical Chat corpus. Each conversation in the dataset is aligned with a topic and several articles of text relating to that topic. An early topic model in Bernard attempted to classify facts into topics, to provide a more robust manner of detection than a naive approach using noun phrases. Here, we trained the model to classify a snippet of factual / fun fact text from Topical Chat into its associated entity (among the 300 present in the dataset). In production, our bot uses nearest-neighbor and TF-IDF metrics to select candidate news stories and facts from our knowledge base given a user statement. Our topic model then predicts the topic cluster for this information, and retrieves additional related supporting information.

While the first iteration of this topic model was trained exclusively on Topical Chat, we found that the limited entity space resulted in unintuitive topic classes (e.g. 'Montana' mapping to 'Georgia'). We thus expanded our training corpus to include a curated set of 20,000 entities available from ASK during the initial setup of cobot, as well as 2,000 additional entities from Reddit threads.

**Topical Chat for BART** We fine tuned the pre-trained BART [15] model on the full TopicalChat dataset. We chose the BART model because it is a pre-trained encoder-decoder architecture, which we find is more appropriate for encoding grounded knowledge that the baseline fine-tuned GPT architecture mentioned in the original TopicalChat paper [9]. In this setting, the encoder can focus on learning how to map knowledge grounding, while the decoder can focus on learning how to produce conversation.

## 5 Additional Datasets

| Split | # Episodes | # Turns | # Sentences | # Words |
|---|---|---|---|---|
| Train | 18,971 | 364,461 | 994,163 | 17.4 M |
| Dev | 2,371 | 45,502 | 123,861 | 2.2 M |
| Test | 2,372 | 44,776 | 122,088 | 2.1 M |
| **2P** | **23,714** | **454,739** | **1,240,112** | **21.7 M** |
| **Full** | **105,848** | **3,199,856** | **7,455,662** | **126.7 M** |

Table 5: Statistics from two-party (2P) and multi-agent (Full) Interview dataset [17]

| Dataset | Spoken | # Dialogs | # Turns | # Words |
|---|---|---|---|---|
| Reddit [28] | ✗ | 147 M | – | 1,800.0 M |
| DailyDialog [16] | ✗ | 13,118 | 102,979 | 1.4 M |
| TopicalChat [9] | ✗ | 10,784 | 235,434 | 4.1 M |
| CALLHOME [6] | ✓ | 120 | 22,214 | 0.3 M |
| Interview 2P | ✓ | 23,714 | 454,739 | 21.7 M |
| Interview | ✓ | 105,848 | 3,199,856 | 126.7 M |

Table 6: Comparative dialog dataset statistics, with two-party (2P) and full Interview dataset [17]

In addition to data provided through the Alexa Prize, we have collected data from several other sources through Amazon-provided crawlers as well as our own web scrapers and publicly available data releases.

Of particular note is the Interview corpus[17], which we collected via scraping transcripts of NPR news radio programs from 1999 to 2019. We selected a subset of conversations to better focus our dialog modeling task on two-party dialogs similar to the setting of a user speaking to an Alexa-enabled device. These conversations in total comprise 23K radio episodes with 455K turns of dialog. Full statistics are shown in table 5,

We noticed an 80-20 imbalance in terms of question asking and answering between hosts and guests, as well as a significant difference in the total volume of dialog on NPR programs, which is indicative of media dialogs [26]. This notion of differing discourse patterns across roles is highly applicable to Bernard, as we operate in a scenario where the user knows the opposite speaker agent and expects (explained in more depth in our Learning from Alexa section) to be talking to a bot. We model Bernard's dialogs primarily from the perspective of a 'host' who, while engaging with the Alexa user in phatic conversation and smalltalk, takes on the responsibility of guiding the conversation. We found that models trained on interviews are better at asking clarifying and follow-up questions. We further noticed that these models condition well on the dialog history as opposed to other dialog that often cannot remember long span of dialog history hence result in inconsistent responses [21].

We compare the Interview dataset to other large-scale and/or spoken conversation datasets for dialog in Table 6, where we see that this dataset comprises several times more dialog volume than comparable proxies for conversational speech data, though not quite at the scale of the Reddit threads corpus.

We additionally collected 78,000+ thread titles from Reddit for a diverse and human-curated corpus of grounding information. We have noticed that Reddit threads tend to be information-dense and concise both to satisfy length constraints and appeal to a greater web browsing audience. Our scrapers use the Reddit Developer API[11] to collect the front page threads daily for several subreddits with large user-bases[12].

To collect topical information, we looked to the subreddits: Movies, Games, AskScience, Futurology, Science, Sports, and Technology. For collecting more colloquial and fun-fact-style grounding information, we also collected thread titles from LifeProTips, TodayILearned, and YouShouldKnow. Finally, we collected news headlines from the news and current-information-focused subreddits News, Politics, and WorldNews to supplement news article titles crawled from the Washington Post, which we scraped with a provided API key for the competition.

For information about movies, we also collected structured information via the IMDB data API[13].

## 6 Dialog strategies

While Bernard has followed various strategies for dialog, we have settled into two main modes of dialog. When we have confidence in the area or domain of discussion, we apply the **Acknowledge-Retrieve-Reply** framework to generate interesting responses, and otherwise we focus on asking **clarifying questions** to elicit preferences, opinions, and queries about a user.

[11] `https://www.reddit.com/dev/api/`

[12] `https://www.reddit.com/r/<SUBREDDIT>`

[13] `https://www.imdb.com/interfaces/`


Figure 4: Breakdown of a bot response using Acknowledge-Retrieve-Reply

### 6.1 Acknowledge-Retrieve-Reply

This is the primary dialog strategy we apply when speaking about domains in which we have large amounts of information. Here, we build a bot response consisting of an acknowledgement of the conversation thus far, followed by a response that is conditioned on information retrieved based on the last user query.

**Acknowledge** In the beginning of most utterances, we provide an acknowledgement of the user's response. This consists of a short, phrase- to sentence-length snippet that tells the user Bernard has been paying attention. Depending on the available conversation context, this can consist of phatic terms like *That's really great to hear* to content-rich terms like *I also like those kinds of horror movies*. The purpose of this segment is to build trust between Bernard and the user, and to acknowledge that in a conversation each participant contributes and learns from one another.

**Retrieve** When there is a factual aspect to a user utterance (e.g. a factual query), the retrieval model seeks the most relevant correct information to construct a satisfactory response. In the general case the retrieval module also plays an important role—while we have encountered chatbots that naively influence the conversation by suggesting arbitrary facts, Bernard seeks out information that is relevant but novel in context of the current conversation session, thus gradually opening up the conversation topics.

**Reply** The primary purpose of Bernard responses are twofold: 1) to keep the user engaged in conversation, and 2) to guide the user toward areas of conversation in which Bernard has high confidence of its responses. While the notion of confidence can be nebulous, the most explicit measure thereof in Bernard is the likelihood of generated text sequences as dialog responses. We maximize this explicitly through beam search decoding in our neural dialog models. In other response generators, we build the confidence in terms of transition probabilities between states in our domain-specific NFAs. Replies may present the user with facts about Bernard's (manufactured) personality, elicit opinions, and answer questions.

### 6.2 Clarifying Questions

In order to elicit user opinions and build trust, our second dialog strategy focused on asking cogent clarifying questions. To ask reasonable questions, we primarily addressed the issue of topicality: how to ask a question that is consistent with the topics of conversation thus far. For each domain-specific module, we used a different strategy to generate these questions. For example, in Movie Chat, we ask users whether they have watched certain movies, and their opinions on those movies. In the clarification state of our dialog state manager, we confirm with the user which topic they wish to discuss, and ask if Bernard has properly understood the query.

We observed that asking polar (yes/no) questions resulted in short user responses and quick disengagement—as a result, we switched to asking more open-ended and subjective questions, allow-

ing users to express themselves naturally. We preserved polar clarifying questions for low-confidence state transitions—e.g. if a user has not watched a movie, we switch to the Movie Information state to provide facts about the film.

## 7 Lessons from Bernard Conversations

Here we present observations and learning from user feedback and qualitative analysis of sample conversations.

**Unrealistic Persona** Chatbot users generally enter the conversation understanding that they are talking to a socialbot. In the majority of cases, the users are willing to suspend their disbelief in Bernard's bot status (e.g. Bernard is incapable of physically cooking) in the pursuit of a pleasant conversation. However, a vocal minority see a severe degeneration of user experience, growing angry and criticizing the bot in conversation about its status as an inanimate object. We find that while incorporating personal experiences increases diversity and the engaging nature of conversations in general, certain assertions may more frequently trigger user backlash. For example, our user study indicates that travel claims are more likely than any other 'experience' snippet to incur negative feedback. We do note that even when responding negatively, users continue to anthropomorphize and address Bernard directly questioning its ability to travel.

**Combative Users** We also notice the presence of combative or adversarial users, whose actions do not align with the goal of engaging long conversations. A subset of these users seek to 'break' the bot by speaking gibberish or constantly abruptly changing topics and contexts. Another subset merely engage Bernard with vile, sexual and racially abusive comments. It is difficult to maintain fluent conversation with such users, and they almost always (99%+) rate the bot a 1/5. We subscribe to the philosophy that *these violent delights have violent ends*, and therefore spend limited resources addressing this problem, with profanity filters and gentle reminders to engage in topical conversation, primarily treating these situations as a 'lost cause'.

**Repetition** Repetition is a known problem with generative models of text and dialog [10]. As the length of the generated utterance increases, the problem compounds. We notice that repetitions can severely degrade Bernard's performance—a pleasant conversation with a user can quickly result in unhappiness with only one or two repeated phrases, depending on the user's mood. To address this issue, we stochastically sample from the output distribution of our neural generation models, and incorporate template-based generation strategies to force diversity and 'unlikely' responses from the perspective of our language model. We have also noticed that the longer our responses tend, the more frequently a user will request a repeat of the last utterance.

**Structured Dialog** In the early stages of the conversation, we relied heavily on asking users for topics of conversation. While creativity is in human nature, this on-the-spot querying combined with the limited topical diversity at the time led to user confusion and many early exits. Adding guiding, clarifying questions and structure to our conversations via the DSM significantly improved our conversation quality, ratings, and mean durations. We were able to take advantage of the modularity of Bernard to quickly add new dialog modules and domain-specific modules to address patterns we observed across user conversations. For example, we created the Movie Chat state based on user feedback and interest in deeper conversations on that topic.

**Applicability of Acknowledge-Retrieve-Reply** The user reaction to our Acknowledge-Retrieve-Reply framework was overwhelmingly positive—as in a human conversation, users seemed to appreciate being acknowledged by the other participant (Bernard). Retrieving a related fact helped greatly increase diversity and coherence in our generated responses. This also introduces new entities to the conversation, motivating users to talk at length or switch to different but related topics. Accordingly, such conversations went on longer and achieved higher human ratings.

**Catastrophic Failure** Although we implemented fallback strategies to provide default responses, ASR misidentifications or key phrase chomping (e.g. a missed 'stop' command) can lead to catastrophic failure. In some cases such as the skill invocation and stoppage, these issues are more specific to the hosting platform (Alexa Skills Kit / Cobot), and affect user agency. We elected for more

aggressive parsing of user responses and capturing of stop commands to address this, as we noticed that one failure to stop on time can very easily lead to a 4.0-5.0 rated conversation turning into a 1.0 conversation. In other cases, a sufficiently scattered conversation history can lead to degenerate neural dialog generation, with at best nonsensical and at worst provocative replies from Bernard. To address this, we elected to reduce the length of the conversation history available to our neural generators, and truncate our bot's memory as the conversation went on longer. High-friction transition from one topic to the other also resulted in a drop in rating despite the rest of the conversation being engaging.

**The Line Between Human and Robot** While some users react negatively to a perceived 'fake' personality, the other end of the spectrum sees users who treat Bernard as a little *too* human and ask subjective questions fundamentally unanswerable from an automated system. While we observed users increasingly asked for relationship advice, such subjective and situation-grounded questions are unanswerable for a socialbot without additional context. This may be an avenue for future research, but we wonder if socialbots will ever be equipped to respond in such scenarios, or whether these sorts of soul-searching conversations should remain the province of human heart-to-hearts.

## 8 Conclusion

We present a dialog framework and accompanying socialbot implementation, Bernard, capable of holding an engaging, open-domain conversation on a diverse range of topics. In the process, we explore various dialog strategies in context of real-world human-to-human and human-to-bot conversation. We propose a stateful dialog manager which infers state transition probabilities based on user response to navigate the conversation flow, and introduce the Acknowledge-Retrieve-Reply system for response generation to increase diversity and engagingness. We show that finetuning powerful pretrained language models on downstream dialog datasets can act as a coherent topical dialog model which seamlessly works with our Acknowledge-Retrieve-Reply framework. In future work, we hope to further explore the role of situational context in dialog, as well as the elicitation and usage of subjective knowledge and opinions for open-domain conversation. We see an opportunity to apply reinforcement learning to learn state transition probabilities in our DSM, especially with human-in-the-loop for real-time feedback.

## 9 Acknowledgements

We would like to acknowledge all the Alexa Prize Teams for helping us with technical guidance and year-long research funding. Thanks to Shivam L., Kunal, and Astuti S. for helping with the engineering load for topical extraction and domain-specific response generators. We would also like to thank our beta testers, especially Rei M., Sujoy P., Alicia L., Timothy S., Eric H., and David Z. who not only provided critical comments on dialog structure but also gave realistic feedback on overall system performance.

## References

[1] D. Adiwardana, M. Luong, D. R. So, J. Hall, N. Fiedel, R. Thoppilan, Z. Yang, A. Kulshreshtha, G. Nemade, Y. Lu, and Q. V. Le. Towards a human-like open-domain chatbot. *CoRR*, abs/2001.09977, 2020.

[2] S. Arora, Y. Liang, and T. Ma. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[3] B. Athiwaratkun, A. G. Wilson, and A. Anandkumar. Probabilistic fasttext for multi-sense word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1–11, 2018.

[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[5] P. Budzianowski, T. Wen, B. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gasic. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026, 2018.

[6] A. Canavan, D. Graff, and G. Zipperlen. Callhome american english speech. *Linguistic Data Consortium*, 1997.

[7] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[8] M. Ghazvininejad, C. Brockett, M. Chang, B. Dolan, J. Gao, W. Yih, and M. Galley. A knowledge-grounded neural conversation model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5110–5117, 2018.

[9] 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 *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 1891–1895, 2019.

[10] A. Holtzman, J. Buys, M. Forbes, and Y. Choi. The curious case of neural text degeneration. *CoRR*, abs/1904.09751, 2019.

[11] C. J. Hutto and E. Gilbert. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014*, 2014.

[12] C. Khatri, R. Goel, B. Hedayatnia, A. Metanillou, A. Venkatesh, R. Gabriel, and A. Mandal. Contextual topic modeling for dialog systems. In *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*, pages 892–899, 2018.

[13] 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. Stubel, 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. *CoRR*, abs/1812.10757, 2018.

[14] G. Larionov, Z. Kaden, H. V. Dureddy, G. B. T. Kalejaiye, M. Kale, S. P. Potharaju, A. P. Shah, and A. I. Rudnicky. Tartan: A retrieval-based socialbot powered by a dynamic finite-state machine architecture. *CoRR*, abs/1812.01260, 2018.

[15] 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. *CoRR*, abs/1910.13461, 2019.

[16] 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, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 986–995, 2017.

[17] B. P. Majumder, S. Li, J. Ni, and J. J. McAuley. Interview: A large-scale open-source corpus of media dialog. *CoRR*, abs/2004.03090, 2020.

[18] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014.

[19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

[20] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990, 2019.

[21] C. Sankar, S. Subramanian, C. Pal, S. Chandar, and Y. Bengio. Do neural dialog systems use the conversation history effectively? an empirical study. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 32–37. Association for Computational Linguistics, 2019.

[22] A. See, S. Roller, D. Kiela, and J. Weston. What makes a good conversation? how controllable attributes affect human judgments. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1702–1723, 2019.

[23] M. Shibata, T. Nishiguchi, and Y. Tomiura. Dialog system for open-ended conversation using web documents. *Informatica (Slovenia)*, 33(3):277–284, 2009.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[25] J. Weizenbaum. ELIZA - a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, 1966.

[26] E. Weizman. *Positioning in media dialogue: Negotiating roles in the news interview*, volume 3. John Benjamins Publishing, 2008.

[27] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2204–2213, 2018.

[28] Y. Zhang, S. Sun, M. Galley, Y. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. *CoRR*, abs/1911.00536, 2019.