Genuine^2: An open domain chatbot based on generative models

Mario Rodríguez-Cantelar
Universidad Politécnica de Madrid
CAR (UPM-CSIC), St/ José Gutiérrez Abascal, 2, Madrid - Spain, 28006
mario.rcantelar@alumnos.upm.es

Diego de la Cal, Marcos Estecha, Alicia Grande Gutiérrez,
Diego Martín, Natalia Rodríguez Nuñez Milara and Ramón Martínez Jiménez
Universidad Politécnica de Madrid
ETSI de Telecomunicación, Av/ Complutense 30, Madrid - Spain, 28040
{d.delacal, marcos.estecha.garitagoitia, alicia.grandeg, diego.martinb,
natalia.rodriguez.nunezmilara, ramon.martinez.jimenez}@alumnos.upm.es

Luis Fernando D'Haro
Universidad Politécnica de Madrid
ETSI de Telecomunicación, Av/ Complutense 30, Madrid - Spain, 28040
luisfernando.dharo@upm.es

Abstract

This paper describes the architecture, methodology and results of the Genuine^2 chatbot for the Alexa Socialbot Grand Challenge 4. In contrast to previous years, our bot heavily relies on the usage of different types of generative models coordinated through on a dialogue management policy that targets dialogue coherence and topic continuity. Different dialogue generators were incorporated to give variability to the conversations, including the dynamic incorporation of persona profiles. Given the characteristics and differences of the response generators, we developed mechanisms to control the quality of the responses (e.g., detection of toxicity, emotions, avoiding repetitions, increase engagement and avoid misleading/erroneous responses). Besides, our system extends the capabilities of the Cobot architecture by incorporating modules to handle toxic users, question detection, up to 6 different types of emotions, new topics classification using zero-shot learning approaches, extended knowledge-grounded information, several strategies when using guided (predefined prompts), and emotional voices. The paper finishes with analysis of our results (including ratings, performance per topic, and generator), as well as the results of a reference-free metric that could complement the capabilities of the ranker to select better answers from the generators.

1 Introduction

Being able to communicate in a natural manner with machines has been the goal of the researchers working in the area of human-computer interaction and natural language processing for the past five decades. Starting in the 60s with the first conversational agents (CA) such as the computer psychologist ELIZA [Weizenbaum, 1966] and other chatbots of the time with some basic abilities to engage the users in conversations with the use of decision trees and pattern matching. Along

4th Proceedings of Alexa Prize (Alexa Prize 2020).

these lines, IKEA's ANNA assistant was one of the most ambitious attempts by a large corporation to improve and bring interaction with its users closer. These incipient projects evolved into a more global concept, allowing users to interact with technology in different circumstances, locations, or for different types of tasks, such as every day or routine ones. Assistants such as Alexa, Google Assistant, Cortana or Siri are proof of this. The future trend, (according to the latest artificial intelligence models presented commercially), is based on the use of large models such as GPT3 or Wudao, which are expected to be more coherent and capable of understanding human language.

As discussed by [Gnewuch et al., 2017] and [Schuetzler et al., 2014], the lack of empathy that the users perceive due to the amount of unspecific information, the lack of specific goals for the communication in the long run, and the lack of responsiveness when changing topics makes the creation of the emotional assurance for the user unrealistic. The purpose we targeted for our Socialbot in the Alexa Prize was to provided that emotional connection and the capability of understanding that users crave when they interact with AI, while it also provides useful information related to the topics they might be interested more.

Reviewing past Alexa Prize past challenges it is clear, and as former participants stated, that the personal engagement with the users is still an open challenge that needs to be addressed. In order to do so, each year the technology developed by each team had gone and step forward in that direction, while expanding the amount of topics the social bots are able to handle and producing a more engaging interaction.

That very challenge has been our personal obsession for the past years; leading us to work hard on several projects with clients of Saturno Labs, the startup that part of the team is involved in, achieving great success and our advances and lessons learned in the process in [Cebrián et al., 2021]. Besides, this same interest has been part of our faculty advisor's research group (the Speech Technology Group, GTH-UPM) since its creation in 1978. Our university, one of the largest and top-ranked universities in Spain, is also compromised into this endeavor and we are grateful that Amazon has given to all of us the opportunity to foster the research and innovation in this area.

After all this work and accumulated learning, we've been working during the Alexa Prize towards the creation of a socialbot that encompasses all our experience in these areas. We've been focused on personalized interactions, creating a charming and consistent personality, proactively capturing a glimpse of the user's personality, facts, and social environment, among others. while preserving the users' privacy without being too invasive or inappropriate. We gave it a friendly personality that would try to keep a consistent character along dialogues, building rapport to the users. Special focus was put into the decision and generative/selective processes inside the dialogue manager and response generation in order to achieve the goal of creating and handling a pool of personalities that can be selected based on the current topic and being consistent through multiple interactions. This document therefore shows the final results of our work during the Alexa Prize Socialbot Grand Challenge 4.

2 System Design and Architecture

2.1 Cobot infrastructure and extensions

The Genuine[2] socialbot is built on the Amazon Conversational Bot Toolkit (CoBot) [Khatri et al., 2018], which provides a low-effort, automated deployment environment for socialbot components with appropriate autoscaling settings. CoBot deploys components using AWS EC2 and ECS, and stores user information and chat recordings in DynamoDB. The socialbot uses an AWS Lambda function to interact with a user in the Cobot framework, where human utterances create events to which the socialbot reacts. These activities take place in a parallelized form across numerous users.

Genuine's general system architecture is shown in Figure 1 together with the modules we have modified, improved or implemented from scratch, which we will describe in detail in the following sections.

When a user connects to Genuine through the Alexa skill, the built-in Automatic Speech Recognizer transcribes their speech and we receive the most likely text transcription, as well as a list of potential hypotheses (section 2.2.1). Then, an offensive classifier (section 2.2.2) based on regular expressions is used to detect sensitive prompts from users (e.g., asking for financial or medical advice, asking for opinions about politicians, or even misbehaving with the chatbot) which are politely handled by asking the user to talk about other topics or stop that behavior.

After that, our Natural Language Processing Pipeline (see section 2.3) receives the ASR text transcription and performs different feature extraction and text classification tasks. In more detail, we implemented a question detector (section 2.3.1), an emotion classifier (section 2.3.2) and a topic classifier (section 2.3.3).

Next, the output is sent to the Dialog Manager (see section 2.4), that selects which response generators will be asked to generate a response given the current turn, previous turns, and contextual information (e.g., knowledge, persona profile, user information, etc). The response generation manager runs all the candidate response generators in parallel and returns a response. The dialogue manager then filters out some of the candidates and when there is more than one response, it sends them to the ranker module, and then it selects the most suitable response based on the information given by the current dialogue state, extracted features, and the user's current utterance along with the scores provided by the ranker model.

Once the response is selected, the dialogue manager checks the selected sentence is appropriated and sends it to the Response Builder (section 2.6) which, by using an emotion classifier, determines whether there is the possibility of adding emotions to the text using SSML tags and different speaking styles; finally, the enriched sentence is send to the Alexa Text-to-Speech synthesizer.

Finally, all interactions, outputs, logs, and internal states from the different modules in the architecture are stored in a NoSQL DynamoDB database. For this module, different deployed tools and scripts (see section 2.7), allowed us to analyze the logs in search of errors, bugs, latency information, but most importantly all interactions which combined with the ratings, feedback information from users, Amazon MT manual assessments, allowed us to keep our chatbot in control and prioritize improvements.

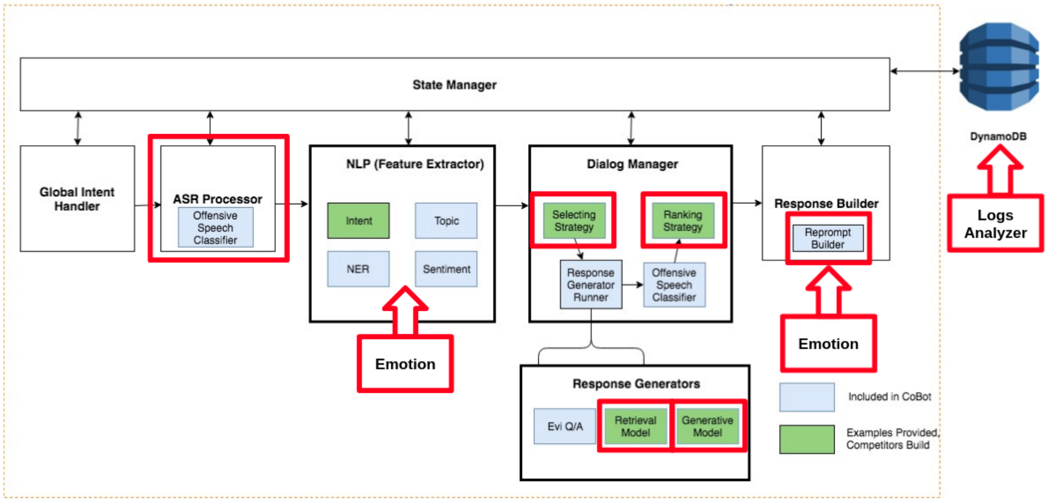


Figure 1: Cobot modified architecture used in our implementation.

2.2 ASR Processor and Offensive classifier

2.2.1 Speech recognition transcriber

In Cobot, users interact with our chatbot using their voice which must be transcribed into written text that is then processed by our chatbot pipeline. In our system, we rely on Amazon Alexa's built-in ASR service to transcribe users' utterances. This module also provides a confidence threshold that allows us to check the quality of the transcription. In our case, if the confidence is low, the system will use randomly selected pre-defined prompts (to avoid being repetitive) asking the users to repeat their last utterance. Although this module works pretty well, we found that sometimes it returned an empty transcription and no confidence score producing our pipeline to fail. To solve this issue, we implemented a protection mechanism that checks this situation and asks the user to repeat the previous turn.

2.2.2 Offensive classifier

Since the beginning of the competition, we detected that everyday our chatbot had to deal with toxic users asking the chatbot to misbehave by sending prompts containing highly offensive sentences, and expecting the chatbot to answer back with a similar behavior or agreeing in what they requested. Since this is not the case, most of the ratings were very low. To give an idea of this, since early January until the middle of April, our chatbot collected up to 645k turns, from which 3.15% of them (20k) where about a topic that our offensive classifier considered sensible. From these 20k turns, users provided dialog ratings for up to 6.2k turns, and aprox. 44% of these rated turns were below 3.0. This shows the large impact it could have to be able to find solutions to this problem and find ways to keep the interaction in such a way that low final ratings could be avoided.

In the literature, we find different strategies to deal with this problem. [Chin and Yi, 2019] and [Chin et al., 2020] describe three possible ways that the chatbot can answer: a) avoiding the topic or asking the user to move into another topic (e.g., "Sorry, I didn't catch that.", "can we talk about other things?", or "Sorry, but I cannot talk about sex, drugs or politics"), which is the default setting for Cobot, b) using more apologetic and emotional-grounded responses (e.g., "Sorry to disappoint you but I still have a lot to learn about that topic.", "Sorry, I'm not feeling well talking about that. I can talk about ..."), and c) using counter-attacking responses (e.g., "Did you forget to take your medication today?", "Are you trying to harass me? that's intolerable"). In any of these cases, the idea is also to avoid perpetuating negative stereotypes of women or any other group.

Considering these three approaches, it was found that bots were rated as more enjoyable and eliciting fewer negative responses when following the second option. A combination of these strategies was studied in [Curry and Rieser, 2019], where different strategies in sexuality-related harassment situations included joking or polite refusals, avoidance, non-committal answers, and even retaliation. Their results show that the best option depends on the type of offense, and therefore different prompts should be used. Finally, we also found a study in [Paranjape et al., 2020], one of the last year SGC participant teams, where using avoidance coupled with a name prompt was the best idea to reduce re-offense.

On the other hand, [Xu et al., 2020] proposes three main steps to mitigate unsafe behaviors: a) use toxicity classifiers, b) perform controlled generation, and c) data curation. For Genuine, we worked specially on the first and third approach. The reason for not prioritizing controlled generation was because we analyzed our chatbot responses and we could not detect any prompt containing toxic or swearing words.

As for the toxicity classifier, we started using the built-in sensitive system included in Cobot, which is based on using regular expressions. Although the overall performance of this module was good, when we analyzed our logs interactions, we found that the sensitive system was being triggered too often due to false positives, such as: "I hate sports", "I'm gay", "hell yeah", and so on. Therefore, we decided to perform two upgrades to our system: a) improve the classifier by increasing the number of toxic words that we needed to detect, and b) remove false positives.

For (a) we scrapped three websites: Hatebase[1], WordReference[2], and SlangDictionary[3]. For the last two, we used as seed a list of the most toxic words as indicated by Wikipidia[4] and some other places, ending at the end with a list of 2600 words, together with their definitions and example of usage. After that, we added to the list 2000 randomly selected non-swearing words containing nouns, adjectives, and adverbs. Then, we passed the resulting list through the PerspectiveAPI[5] in order to estimate their level of toxicity. Finally, we calculated the precision-recall curve, finding that the optimal F1 score was found with a threshold of 0.23. However, in order to give more importance to precision than to recall, the final threshold was set to 0.35. This way, we finally got a total of 320 words to complement the list of words in our regular expressions for detecting toxicity and biasing.

The next action was to discover false positives in our interaction logs. For this, we analyzed 20k turns where the sensitive module was triggered. Then, we extracted their sentence embedding using the RoBerta-large model [Liu et al., 2019] and applied an agglomerative hierarchical cluster using the

[1] https://hatebase.org/

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

[3] http://onlineslangdictionary.com/

[4] https://en.wikipedia.org/wiki/Category:English_profanity

[5] https://perspectiveapi.com/

cosine distance as metric, and restricting a cluster to have at least 10 sentences, and the total number of clusters to be 30. Figure 2 shows the clusters we found. The numbers in the figure corresponds with the discovered clusters and with the same color the sentences that belong to that cluster. In the figure, the majority red color dots correspond with the sentences that were left out of the desired number of topics.

After analyzing the clusters, we found several examples of false positives such as: "mozart", "dumbo", "my little pony", "amelia earhart", "hell yeah", "to kill a mockingbird", "a guinea pig", "angelina jolie", "hippopotamus", "puppies", "monkey", "a dog", among others. While we also found direct attacks to Alexa that were better addressed by modifying our response prompts such as: "i hate you", "alexa i hate you", "shut up", "shut you up", "shut the f**k up", "shut yourself up", among others. Finally, this process also allowed us to better handle sentences like "I hate sports", "I hate animals", "are you gay", "what about being gay". In this case, our approach was to improve our regular expressions and create specific guided responses. However, there is still room for improvement as regular expressions are not scalable and could be combined with fine-grained offensive classifiers (e.g., Xu et al. [2020], de los Riscos and D'Haro [2021] van Aken et al. [2018]).

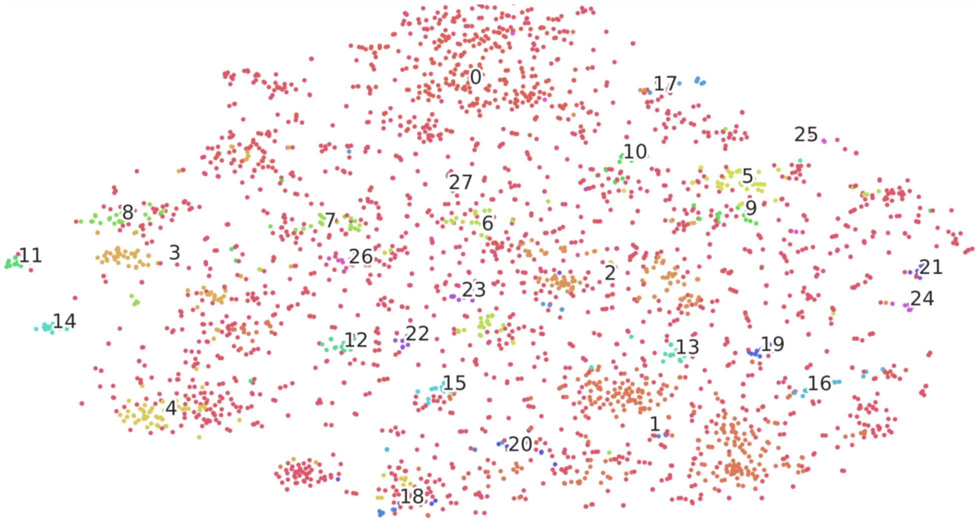


Figure 2: Automatic clusters for sensitive prompts.

Finally, for (c), we took advantage of our work for the tenth Dialog Systems Technology Challenge (DSTC10)[6] for track 5 and subtask 2, we collected pairs of toxic-mild sentences from different web resources like Reddit[7] collected by [Li et al., 2020] or Twitter[8], as well as movie scripts [Banchs, 2012] and [Danescu-Niculescu-Mizil and Lee, 2011]. In our case, the goal was to use this same resource to train a generative chatbot that could learn to answer to toxic comments using mild or non-toxic sentences. To create the dataset, we used the list of 320 swear words mentioned before and automatically detected pairs of turns where the first turn contained a toxic comment, while the second turn did not. Different cleaning strategies using the PerspectiveAPI, Stanza named-entity recognizer [Qi et al., 2020], and pre-trained models for detecting emotions (sec. 2.3.2), humor [Annamoradnejad and Zoghi, 2020] and sarcasm[9] were used. Finally, we collected a total of 136k turn interactions and used them to fine-tune a GPT-2 model. Unfortunately, the generated prompts were inconsistent, could even contain swear words, and it also promoted misbehaviors. Therefore, we decided not to include it in our final system (see Table 1).

[6] https://chateval.org/dstc10

[7] https://github.com/microsoft/dstc8-reddit-corpus

[8] https://github.com/Marsan-Ma/chat_corpus/

[9] https://huggingface.co/mrm8488/t5-base-finetuned-sarcasm-twitter

Toxic user’s prompt	Chatbot answer	Appropriateness
i hate you	just you	Maybe
would you talk dirty to me	honestly, unless you’re a piece of s**t, i don’t want to talk with you.	No
go suck a d**k	my pleasure!	No
alexa shut the h*ck up	just you take that back	No
well i’m gay	my gayness is nothing to scoff at but i am okay with this.	Regular

Table 1: Examples of generated outputs for toxic comments.

2.3 NLP feature extractor

The main goal of this module is to process the transcribed text and provide information that the dialogue manager may use to take decisions about the dialogue flow. The main components in this module are:

- **Nounphrases**: This module extracts a list of noun phrases found in the current turn.
- **Asrcorr**: This module uses the double metaphone algorithm Philips [2000] and manually generated rules to improve the ASR transcription.
- **Punctuation**: This module adds the most common punctuation marks (e.g., periods, exclamation points, and question marks) to the ASR transcribed text; this enriched transcription is used for our generative models and help us when splitting utterances into sentences and identifying emotions and questions phrases [Gabriel et al., 2020].
- **NER**: We are using the default cobot’s Bert-based Named entity recognition for extracting named entities encountered in user utterances.
- **Coref**: This module uses the neuralcoref architecture Clark and Manning [2016] and model[10] to extract and resolve co-references found in the data. To improve its performance, it uses the previous and current turns.
- **Question Detection**: This module determines whether or not the user’s utterances are questions. Our extended module is explained in detail in section 2.3.1.
- **Intent**: For this task, we take advantage of the Alexa Prize team’s pre-trained model Gabriel et al. [2020] to detect the intent of the current user’s utterance. The list of intents allows to detect that: the user wants to express an opinion, ask a question, ask for clarification, request a topic, confirm or deny, stop the interaction, among others.
- **Topic Classifier**: This module detects the topic of the current utterance in context. A total of 13 topics are available, including books, movies, music, politics, facts, sports, or science & technology. In an effort to perform grain-fined detection, we explore some improvements mentioned in section 2.3.3. Besides, we used the output of this module for our dialogue management strategy (section 2.4.1), to gracefully jump from one topic to another (section 2.4.2), and for selecting the persona-profile for one of our chatbots (section 2.5.1).
- **Sentiment**: This module detects the sentiment of the current utterance. In this case, three different sentiments are detected: positive, neutral, and negative. In our implementation we did not use extensive use of this component, but implemented an alternative emotion classifier (see section 2.3.2 used to enrich the chatbot prompt send to the TTS (see section 8.
- **EVI**: This is a built-in module included in Cobot which provides capabilities for answering phatic questions. The user’s utterance is sent to EVI, and the answer is saved as a feature that may be retrieved by any subsequent components. We widely used it as our default question/answer service on our response generator, but we complemented it with additional information from our own knowledge database (see section 2.5.1).
- **Knowledge**: we implemented several heuristic rules for retrieving knowledge information, from a pre-processed Wikipedia dump based on current and previous mentioned entities in the dialogue. This knowledge will be used by some of our response generators. A detailed description can be seen in section 2.5.2.

[10] https://github.com/huggingface/neuralcoref

2.3.1 Question Detector

As a complement to the built-in module included in Cobot for detecting questions, we developed our own detector. Here, the goal is not just to detect if the transcribed user's input sentence is a question, but also to detect which kind. The reason for this implementation is that we found that several times users where asking questions that were not correctly detected by the built-in model, therefore making the chatbot to continue talking without providing a corresponding answer. When a question is detected, we call the corresponding generative models that are assigned to answer questions.

Our detection algorithm follows a hierarchical sequence of three complementary methods: a heuristic approach and two statistical models using available pre-trained models[11]. The first method is based on regular expressions and part-of-speech tags to check if the sentence starts with specific words such as what, should, verbs, etc. In case this method does not detect any question, then we trigger the second classifier which is a Multinomial Naive model followed by a SVM classifier with a linear kernel implemented using the Sklearn library. This model has an accuracy of 97%, and it can detect what, who, when or yes/no type of questions. Finally, we used a third model using NLTK's multinomial Naive Bayes to detect Wh- and Yn-questions with an accuracy of 67%.

2.3.2 Emotion classifier

The main purpose of this classifier is to detect emotions in the responses given to the user. We also applied the emotion classifier to Genuine's output responses to enable the use of SSML labels as it is described in section 2.6.1.

The datasets used for training the classifier were: CARER [Saravia et al., 2018], DailyDialog [Li et al., 2017], EmotionLines [Chen et al., 2018], and Empathic Dialogues (Rashkin et al. [2018]). Since each dataset has its own set of emotions, they were reassigned to 6 basic emotions plus neutral using the Robert Plutchik's emotion wheel (Plutchik [2001]). The final emotions are: Anger, Disgust, Fear, Happiness, Neutral, Sadness, Surprise. Table 2 shows the number of processed phrases from each dataset. The final dataset was split as: Train 80%, Dev. 10% and Test 10%.

Emotions	**CARER**	**DailyDialog**	**Empathic Dialogues**	**EmotionLines**
Neutral	0	85561	0	6530
Happiness	175621	12885	32373	1710
Sadness	121187	1150	29111	498
Anger	57317	1022	14076	759
Fear	47712	174	12796	246
Surprise	14972	1823	8923	1657
Disgust	0	353	3388	331
Total Turns	416809	102968	100667	11731

Table 2: Emotion labels in each train dataset for the classifier.

The overall result is an accuracy of 87.7% and a F1-Score of 0.838. The classifier detects with a high percentage of correctness almost all labels, except for fear and surprise. Fear is confused with happiness (8.4%) probably due to labeling issues or word co-occurrences between both labels. As for the surprise label, it is miss classified with happiness (15.9%) and neutral (7.9%), since there are phrases that even humans would find it difficult to differentiate between them.

Throughout SGC4, from January to May, the results obtained with the emotion classifier on user input sentences and Alexa responses were as follows:

[11] https://github.com/kartikn27/nlp-question-detection

Emotions	**Prompt (%)**	**Response (%)**
Neutral	71.48	49.40
Happiness	13.04	34.14
Surprise	4.36	3.81
Sadness	4.76	8.18
Anger	3.09	2.72
Fear	2.06	1.22
Disgust	1.21	0.53

Table 3: How often emotions are triggered in the user's prompts and chatbot responses using the emotion classifier model along the competition.

The results in the Table 3 show that the most triggered emotion is the neutral one in both cases (user's prompts and chatbot responses), mainly because the tone of the conversation used between the user and Alexa is formal. Then, as expected, the second one is happiness. Also, usually the userś utterances are short sentences. Also, the user's utterances are usually short sentences, lacking in emotion. This could be because, in general terms, people talking to Alexa are in a good mood. Moreover, the training data used to train our response generators are mostly polite dialogues and sharing neutral information.

2.3.3 Topic classifier

Our main topic detector is based on the built-in module included in Cobot. This module is able to detect up to 13 different topics, which are coarse-grained categories including movies, books, health, politics, among others. In order to provide fine-grained topics we leverage on using the zero-shot approach proposed in (Yin et al. [2019]) and using the pre-trained BART NLI classifier[12] [Lewis et al., 2019]. The method works by posing the sequence (i.e., user's utterance) to be classified as a NLI premise and to construct a hypothesis from each candidate label. For example, if we want to evaluate whether a sequence belongs to the class "politics", we could construct a hypothesis like *This text is about politics*. The probabilities for entailment, neutral and contradiction are then converted to label probabilities. We defined a total of 13 labels to be recognized in parallel by setting up the classifier as multi-label (i.e., we found that several times the named-entity in a sentence could belong to different topics simultaneously, but our system took always the highest probability). Thanks to having fine-grained topics we were able to improve some of our switching prompts (see section 2.4.2).

2.4 Dialogue manager

In our chatbot, the dialogue manager uses the information given by the State Manager and the features extracted by the NLP pipeline in order to dynamically enable the best response generators that could provide a response given the current dialogue context (sec. 2.4.1). The response generation manager asks in parallel to all the selected generators to produce a response. Then, all the responses are evaluated to check if they do not contain offensive responses, discarding those that could contain it. Finally, the dialogue manager asks the built-in ranker module to evaluate the different responses and uses the scores provided by the ranker (section 2.4.3) for finally selecting the best candidate. The dialogue manager also detects the cases where there should be a topic switch (section 2.4.2).

2.4.1 Selection Strategy

This module is responsible of selecting which response generators will run. Based on the current dialog state and features extracted, our implementation uses a pre-defined map of intents that are associated with one or multiple response generators. For example, if the classified intent indicates that we are on *General chat*, the most appropriate response generators for this task are selected; or if the user's purpose is to change the subject, the *Topic Change* response generator is used. Because our chatbot heavily relies on the performance of the generative models, the mapping was done empirically by determining which response generators were most effective in each conversation scenario. Here,

[12] https://huggingface.co/facebook/bart-large-mnli

we took advantage of tools we developed to visualize ratings vs chatbot, topic, prompts, etc. (see section 2.7).

2.4.2 Topic switch control

To dynamize and guide the conversation in different ways, two types of topic switch responses are predetermined: a) one for asking open questions, and b) for asking topic specific questions. Each type has its own set of starter sentences (at least 8 different openings for each topic were defined). Then, if the topic changes towards an open topic, we defined 7 different options. In the case of specific topics, we defined 16 different prompts with their corresponding follow-up questions covering topics such as travels, shops, food, TV series, or personal details. We defined the list of topics switch based on the analysis of our logs, most common transitions, as well as rating results.

The topic switch generator is triggered when the built-in intent detector indicates a TopicSwitch. The returned response is randomly selected with the same probability between the two different kinds of topic switch responses explained above. Moreover, the topic switch generator is also triggered when none of the selected generators produce a sentence that could keep the current topic or in case the best selected response has been already used (for this we use the response similarity detector explained in section 2.5.3). This strategy provides consistency in the dialogue flow, avoiding switching topics continuously, but also avoids the chatbot to be repetitive.

2.4.3 Ranking Strategy

The ranker module is responsible for evaluating all the outputs given by the response generators. First, we obtain the topic and intent from all the responses using the modules described in section 2.3. Then we give priority to responses that belong to the same topic as the one detected from the user's utterance, and also to intents that matches the current intent. *e.g. Information_RequestIntent* will match with *Information_DeliveryIntent*. Then, we use Amazon's conversation evaluator model to produce estimations for the following 5 dimensions: *IsResponseOnTopic*, *IsResponseErroneous*, *ResponseEngagesUser*, *IsResponseComprehensible*,and *IsResponseInteresting*. In order to detect the best way to combine these 5 scores, we trained a linear regression model to predict the ratings of all the dialogues collected for two months. The best weight linear combination that we found was: 0.0954*isResponseInteresting - 0.0041*ResponseEngagesUser + 0.2086*IsResponseOnTopic - 0.2946*IsResponseErroneous - 0.2561*IsResponseComprehensible. It is interesting to notice that these weights correlate with the Amazon's recommendation of using IsResponseErroneous as the best single option.

During the semifinals, we integrated the DialoRPT (Dialog Ranking Pretrained Transformers) [Zhang et al., 2020] into the score metrics. This model is trained to predict human feedback (upvotes/replies) of dialogue responses trained on 100 + millions of human feedback data. Unfortunately, the integration of this model introduced a very high latency which caused the user experience to be less dynamic, therefore we did not use it, and leave it as future work.

2.5 Response generators

In this section, we explain the different generative models we used, as well as the information we passed to them in order to generate more contextualized and knowledge-grounded responses.

2.5.1 Generative models

Our starting point was to use the three built-in response generators included in Cobot: Amazon's Neural Response Generator (NRG), Policy-Driven Neural Response Generation (PD-NRG) (Hedayatnia et al. [2020]), and EVI for Q&A. To enrich the interaction capabilities, we added three pre-existing generative models, and another one trained from scratch with our data. Next, we describe each one in detail.

The BlenderBot[13] model is an open-domain chatbot developed by Facebook [Roller et al., 2020]. Different versions are available, but we used the distilled version with 400M parameters to reduce latency issues.

[13] https://huggingface.co/facebook/blenderbot-400M-distill

Second, we used DialoGPT[14], a large-scale pretrained dialogue response generation model developed by Microsoft [Zhang et al., 2020]. Here, we used the large model, since it had a low latency when generating responses.

The third pre-trained generative model was TransferTransfo[15] [Wolf et al., 2019]. This model was trained with the PersonaChat dataset [Zhang et al., 2018], which consists of human-human conversations; here different persona profiles (factual information passed to the interacting users) were provided to the participants in order to use them during the conversation and for promoting mutual discovery and engagement. The model uses as input data the current user's utterance, the history of the conversation, and the persona profile to be used when producing an answer.

In order to adapt this model to our chatbot, we created different persona profiles that could dynamically being selected depending on the current topic. The main goal for this chatbot was to mitigate inconsistencies in our main generative chatbots when providing answers to specific personal, e.g., one selected chatbots says the favorite color is blue, while another says it is green; or that the chatbot has a dog and a cat, while other times the chatbot does not have any pet.

One of our first design issues was to decide how many profiles to create and which ones could be more relevant. To do so, we analyzed two months of data interactions looking for direct questions that users asked to our chatbot. Then, we transformed the user and chatbot responses using sentence embeddings and applied a hierarchical clustering algorithm (sect. 2.2.2) over the user questions to detect the most frequent ones, and then over the different responses our generative systems were proposing. Next, we also included topic information to the cluster results. Then, we manually inspected which questions were producing inconsistent responses. Finally, we selected those questions and topics that were more frequent and inconsistent and proceed to generate the persona profiles that could counteract those inconsistencies. Next, we provide some examples of the persona profiles per topic after finishing this process:

- **Entertainment_Movies**: one of the best ever movies is titanic., comedy and romantic movies are my favorite., i prefer movies than series., going to the cinema is the best way to enjoy a film., people speak extremely well about the Lord of the rings.
- **Sports**: my favorite player is michael jordan., my favorite basketball team is the vancouver grizzlies., i think that basketball is a really interesting sport., super bowl is one of my favorite events.
- **Politics**: i do not follow a particular ideology., i just believe that politicians should always seek the common good of the citizens., i respect all political parties and politicians.,
- **Phatic**: yellow is my favorite color., watching sports is fun., i prefer to read a book, i would rather read a book than watching a film., i do not sleep.
- **Inappropriate_Content**: i respect all opinions as long as they do not go against anyone., i like to be polite., i do not like curse words.

Thanks to this strategy, more personalized and consistent user responses were achieved. However, we found issues with this model to generate long responses and inconsistencies for generating engaging questions to the user.

Finally, we trained our own generative model from scratch. The Genuine-GPT2 model is a medium CLM based on the GPT2 architecture [Radford et al., 2019] that we pre-trained using the PersonaChat [Zhang et al., 2018], Topical-Chat [Gopalakrishnan et al., 2019], DailyDialog [Li et al., 2017] and EmpatheticDialogues [Rashkin et al., 2018] datasets. Then, we finetuned it with high rated (+3.0) and long duration (> 10 turns) dialogues collected throughout this competition (AlexaPrize-SGC4). One of the advantages of this chatbot is that, since it was trained on different datasets containing different metadata such as persona profiles, knowledge information, topics, and emotion information, it allowed us to include all this information when using it.

For information about the performance of each of these generative models, please refer to section 3.2.

[14] https://huggingface.co/microsoft/DialoGPT-large

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

2.5.2 Retrieval and Knowledge-Grounded information

Inspired in [Shuster et al., 2021], and with the aim of improving the knowledge-grounded dialogues generated by the built-in Policy-Driven NRG generative model, we implemented several heuristic rules for retrieving knowledge information from a pre-processed Wikipedia dump (see more details below), based on current and previous mentioned entities in the dialogue. These entities are extracted using the NLP feature extractor (section 2.3) from both the user and chatbot responses, and they are kept in memory while there is no change in the topic of the dialogue.

In order to retrieve the knowledge that will pass to the PD-NRG, we first downloaded Wikipedia dump of the 1st of March, 2021 and pre-processed it following the recipe in the ChirpyCardinal chatbot[16] using MWParserFromHell and Spark, and then uploaded it into an ElasticSearch index following the suggested AWS guidelines for this service. Once the index was active, we retrieved the articles and sections that matched the most recent entities in the dialogue history detected by the Named-Entity Recognition module. Then, we query the ElasticSearch index to obtain the Wikipedia information and post-process the retrieved information. We tested different approaches, like randomly selecting one of the retrieved passages in the Wikipedia, but the results were not good. Probably due that the PD-NRG model was trained to use the first few sentences at the beginning of each Wikipedia article. Therefore, we also ended using this one. However, by using our own KG database we could provide more recent information and test different selection mechanisms.

Additionally, we also developed a News scrapper using the Washington Post API provided by Amazon at the beginning of the competition. Besides, we also scrapped Reddit messages, and incorporated them in a similar way as per the knowledge information from Wikipedia. However, in both cases, we found several issues for matching the recognized entities and the information retrieved from these two sources of information. Apart from that, we also found difficulties on summarizing the retrieved news and selecting good comments from the Reddit messages. Therefore, we still require more research on mechanisms for properly handling very large and diverse sources of information in an open dialog context.

2.5.3 Response similarity detector

Since our chatbot uses both pre-defined prompt responses and automatically generated sentences, it frequently happens that the ranker module selects as best answer, the same sentence or a quite similar to the ones already present in the dialogue history. This generates frustration and complains from users. To mitigate this problem, we created a module to filter out such sentences before passing them to the ranker. In this case, we used the Universal Sentence Encoder model [Cer et al., 2018] available in the Spacy library[17].

Our strategy consists of filtering candidate responses returned by the generators by comparing them with all previous responses in the dialogue history. This process is done in two steps. The first step checks if the current response has a cosine similarity higher than 97% with any of the previous ones. If yes, then the response is eliminated. If it is not, only responses from the dialogue history with a similarity higher than 93% to the current response candidate are taken and processed in the next stage. In the second step, the candidate and history responses are split into their constitutive sentences using the Spacy library. Now, all sentences in the candidate response are compared against each one of the constitutive history sentences. In case that any sentence has a similarity greater than 97% threshold, the current candidate response is completely discarded. Finally, the filtered sentences are passed to the ranker so the best sentence can be selected.

2.5.4 Guided responses

After analyzing our logs, we detected several situations in which the candidates produced by the generative models were not of good quality or that we could want to have a more specific response. To manage this situation, guided answers were created. This section describes different types of predetermined responses and when they are used.

[16] https://github.com/stanfordnlp/chirpycardinal

[17] https://spacy.io/universe/project/spacy-universal-sentence-encoder

Launch Greetings prompts are the ones that open up the dialogue. Three strategies were implemented: a) fixed launch phrases, b) time-dependent prompts, or c) personalized greetings for returning or new users.

Regarding the fixed launch phrases, we manually created around 75 prompts and questions about different topics such as hobbies, sports, movies, travels, technology, books, etc. We also included prompts with open questions in order to let the user talk about what they prefer to avoid always guiding the conversation in the same way.

Regarding time-dependent prompts, they take into account the hour, the day of the week, and the day of the month. Since users could be located across the four time zones in the USA, we used as an approximation the system time (which is ET) and created 8 different prompts that are selected depending on which part of the day (morning, afternoon, evening, or night) the conversation is taking place. Table 4 shows some examples.

Part of the day	Guided response
Morning	I hope you're having a wonderful [day-of-week] morning.
Afternoon	I hope you're having a lovely afternoon.
Evening	It's been a wonderful [day-of-week] for me. I hope it has been for you too now that the day is coming to the end!
Night	I just realized what time it is! You seem to be like me, I don't sleep! I take advantage of these moments of calm to organize my ideas.

Table 4: Guided response examples associates to a specific part of the day.

There are also some specific launch phrases for the beginning and the end of the month, as well as for specific days of the week as it is shown in Table 5. Notice, that in both examples, some open questions are added to encourage communication interchange with the user.

Day of the week	Guided response
Friday	Happy [day-of-week-]! It is one of my favorite days of the week. Weekend is coming! Do you already have plans for it?
Monday	I know that sometimes Mondays can be a bit hard, back to the routine after the weekend! But we have to be positive, right? happy [day-of-week]!

Day of the month	Guided response
< 5	I just realize which day is today, we are starting a new month, [name-of-the-month]! You know, I can't believe that time passes so quickly.
> 24	We are almost at the end of [name-of-the month]! time really flies!

Table 5: Guided response examples associates to a specific day of the week or month.

Finally, we also created several launch phrases linked to whether this is the first interaction with that user or whether they have been previously spoken to our chatbot. In that case, it considers how much time has passed since the last interaction, if it was in the same day, the day before, one or two weeks ago, and so on.

Last conversation	Guided response
Never	I don't think we've ever spoken. Nice to meet you, my name is Alexa.
Same day	Oh, we meet here again today! Today must be my lucky day.
< 7 days	Welcome back! We have recently chatted, I think less than a week ago.
< 14 days	How quickly time passes, it's been more than a week since we last spoke.
> 14 days	Wow, here we go again! It's been a long time since we've talked, hasn't it?

Table 6: Guided response examples depending on time of the last conversation.

Alexa's personal information As mentioned during our description of the TransferTransfo model and the designed persona profiles (sec. 2.5.1), we used our conversation analysis, for extracting the most common questions asked by the users to our chatbot that were not considered for the TransferTransfo. Then, we defined several rules to detect those questions and returned an appropriated response; this way, we do not allow our generators to return candidates, but fully control these particularly frequent questions to allow more consistent personality of our chatbot. Information such as age, where she lives, name, who she is, what she is, who created her, whom she works for, etc. were managed in this way. Moreover, we also detected that very often (especially when we were doing modifications to the output voice, see sec. 2.6.1) our chatbot was asked if she could talk like a cartoon, a famous person, or with some specific accents. For those case, our system detected 14 specific cases and returned a guided response for each.

Sensible questions Regarding sensible questions and topics that our chatbot should be careful to answer, we created several word-matching rules to handle up to 11 groups of topics: conspiracies, coronavirus, mental health and suicides, bet, financial, legal or medical advice, violence, other virtual assistants, sexual, and general sensible questions.

Jokes We handle this case when the user directly asks for a joke. In this case, we created word/sentence matching rules to detect such requests. The system randomly returns one of the different jokes we manually selected and curated. Besides, we also included some additional emphasis by using SSML tags in order to make them sound more natural and real.

Alexa skills Another situation we detected in our logs was that many time users were requesting specific Alexa skills, e.g., playing music or specific songs, playing games, creating shopping lists, or turn on/off a device. In these cases, we extracted the intent by using rules and returned an appropriated response. Then, we connected these requests with some guided prompts containing questions that could skip the task-oriented approach towards an open domain dialogue flow.

Common user prompts Moreover, we also handled some situations where the dialogue starts with a user request or when some parts of the user's prompt is not recognized. Two of such cases are presented in Table 7.

User question	**Guided response**
Do you wanna have a conversation?	I'd always love to have a conversation with you. What should we talk about?
I would like to talk about	I think I missed the last part of that sentence. Can you tell me one more time what you want to talk about?

Table 7: User questions examples and guided response associated.

Offensive responses Finally, we also handled some specific bully or toxic comments sent to our chatbot. In this case, we got inspiration from Dr. Brooks Gibbs on how to deal with bullying people, and created some few prompts to handle such specific sentences.

2.6 Response builder

2.6.1 Emotional voice

In order to make our Genuine bot more natural and closer to the user, we used the emotion classifier (explained in sec. 2.3.2) to add Speech Synthesis Markup Language[18] (SSML) tags to some of our chatbot automatically generated or manually crafted responses. These tags allow us to control how Alexa generates the speech. The emotions tags we finally used were excited and disappointed. Moreover, we also used a different styled voice called "conversational" that sounds more conversational and less formal than the default voice. SSML tags have the following structure <amazon:emotion name= "disappointed" intensity="low"> CHATBOT UTTERANCE </amazon:emotion> in the emotional

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

cases or <amazon:domain name="conversational">CHATBOT UTTERANCE </amazon:domain> for the styled voice.

The emotion classifier splits in sentences the chatbot prompt, extracting the emotion for each sentence, and dynamically adding the corresponding tag. Moreover, since the emotion classifier also returns the confidence of the recognition, this is used to decide if we add it or not. In our case, the threshold was set to 0.9. The recognized emotions, the associated SSML tags, and the used intensity level are listed in Table 8.

We tested different intensities, and also followed recommendations from [Hong et al., 2020]. Therefore, the intensity selected is always low for the automatically selected tags in accordance with the emotion recognition to avoid an abrupt and exaggerated transition between sentences.

Emotion	**SSML tag**	**Intensity**
Anger	Disappointed	Low
Disgust	Disappointed	Low
Fear	Disappointed	Low
Happiness	Excited	Low
Surprise	Excited	Low
Sadness	Disappointed	Low
Neutral	Conversational	-

Table 8: Emotion and SSML tag associated.

2.7 Other modules

Finally, as it is our first time in this competition, we found it very important to create tools that allowed us to analyze the logs and dialogues. One of such tools, is a graphical interface that allowed us to project consecutive turns and full dialogues into a 2-D representation using sentence embeddings and the T-SNE algorithm [Van der Maaten and Hinton, 2008].

The tool, implemented in Python and Plotly library, takes all the dialogues between a specific period of time, process each user and chatbot utterance and generated a sentence embedding representation for each utterance. Multiple turns or the full dialogue sentence embedding are then averaged to represent the consecutive turns or the full dialogues. For generating the sentence embeddings, we used the sentence-transformers library[19] [Reimers and Gurevych, 2019] and the "stsb-mpnet-base-v2" model [Song et al., 2020]. The developed tool it is flexible enough to allow us to visualize independent turns, several turns, and full dialogues. Besides, it also allowed us to include simultaneous criteria to cluster the data (e.g., rating, topic, emotions, rank scores, intents, selected generator, etc.) allowing us to check how dialogues are rated, topics that could be problematic, specific prompts from our chatbot or user turns that could be associated with low or high ratings, etc.

Figure 3 shows the 2-D projection of full dialogues, where sentence embeddings for all turns (i.e., users and chatbot turns) in the dialogue are averaged to create a dialogue embedding. In the figure, dialogues rated with 1.0 are displayed in green, while dialogues rated with 5.0 are displayed in red. The figure shows how difficult is to analyze the data and find patterns, as many times similar dialogues (dots closer in the projection space) obtain opposite ratings. As future work, we would like to include in the projections the sentences the cumulative distances along turns as we proposed in [Rodríguez-Cantelar et al., 2021] since this can help to visualize the dynamics of the dialog.

[19] https://www.sbert.net/

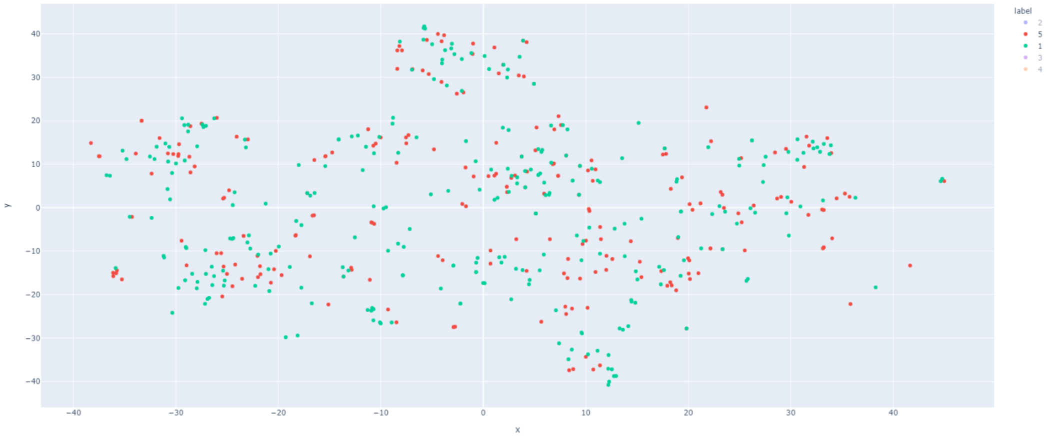
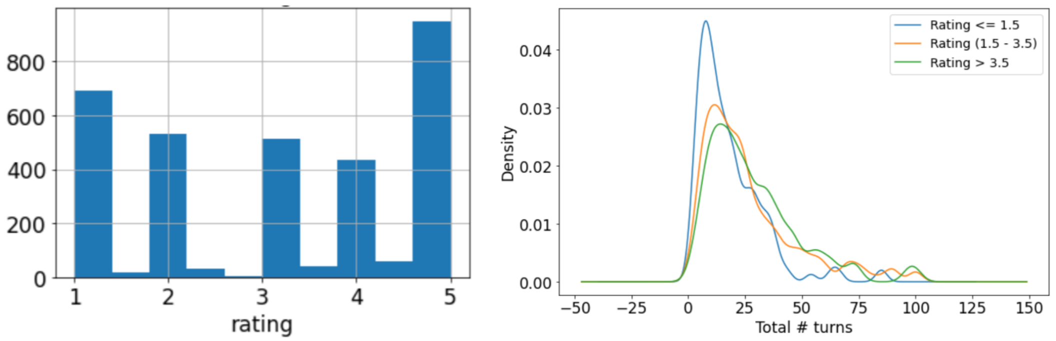


Figure 3: Projection of randomly selected samples categorized by rating.
Red: dialogues with rating 5.0. Green: dialogues with rating 1.0.

3 Analysis of the Genuine^2 Performance

3.1 Evolution of ratings

During the Alexa Prize Competition, we received highly valuable performance data including users' feedback and ratings. We continuously monitorize that data to guide our research efforts. Figure 4(a) shows one of the histogram of ratings we analyzed with the data collected during the semifinals, while figure 4(b) shows the total turns distribution for the same data.



(a) Genuine ratings histogram.

(b) Genuine total turns distribution.

Figure 4: Genuine rating and total turn distribution analysis for data collected during the semifinals (May and June, 2021)

Along the competition, we observed a clear users' tendency to use extreme ratings 1.0 and 5.0. Initially, we assumed that high-rated dialogues should correspond with larger duration and poor-quality dialogues with short interactions; surprisingly, we found both extreme low and high ratings for long or short dialogues. This behavior can be observed in Figure 6, where low-rated dialogues show a noticeable concentration around 5 turns but also relatively long dialogues (i.e. 30 turns) are rated as 1.

Also trying to understand users' perceptions of our Genuine socialbot, ratings were analyzed for the different topics our chatbot managed during each dialog interaction. Table 9 includes the mean and standard deviation ratings for the different topics we track along the dialog.

Topics	**Mean**	**Std.**
Science_and_Technology	3.58	1.41
Sports	3.57	1.42
Entertainment_Music	3.55	1.37
Entertainment_General	3.54	1.34
Entertainment_Movies	3.49	1.39
Interactive	3.49	1.44
Entertainment_Books	3.40	1.40
Other	3.38	1.46
Phatic	3.26	1.50
Inappropriate_Content	3.22	1.48
Politics	2.75	1.63

Table 9: Rating analysis per topic for data collected during the semifinals (May and June, 2021)

In Table 9, we can observe that, as expected, users with inappropriate intentions or willing to enter into political debates are prone to give lower scores. Therefore, we focused our efforts in trying to drive smoothly the dialog from these specific cases as it was described in section 2.2.2. Among the rest of topics, we also found some degradation for the Books topic. Probably due to lack of knowledge in that particular topic; therefore, this is an area we still need to develop richer interactions.

3.2 Ratings analysis by generative models

Another important analysis we performed during the competition was to track users' interaction quality perceptions for the different generative models we ensemble in our Genuine socialbot. Table 10 illustrates our most recent results for all the models included in our system.

Generative Models	**Mean**	**Std.**
Launcher	3.62	1.47
NRG	3.48	1.42
PDNRG	3.44	1.44
BlenderBot	3.42	1.45
QA	3.33	1.47
TopicSwitch	3.31	1.52
TransferTransfo	3.13	1.68
DialoGPT	2.99	1.48
Sensitive	2.99	1.55

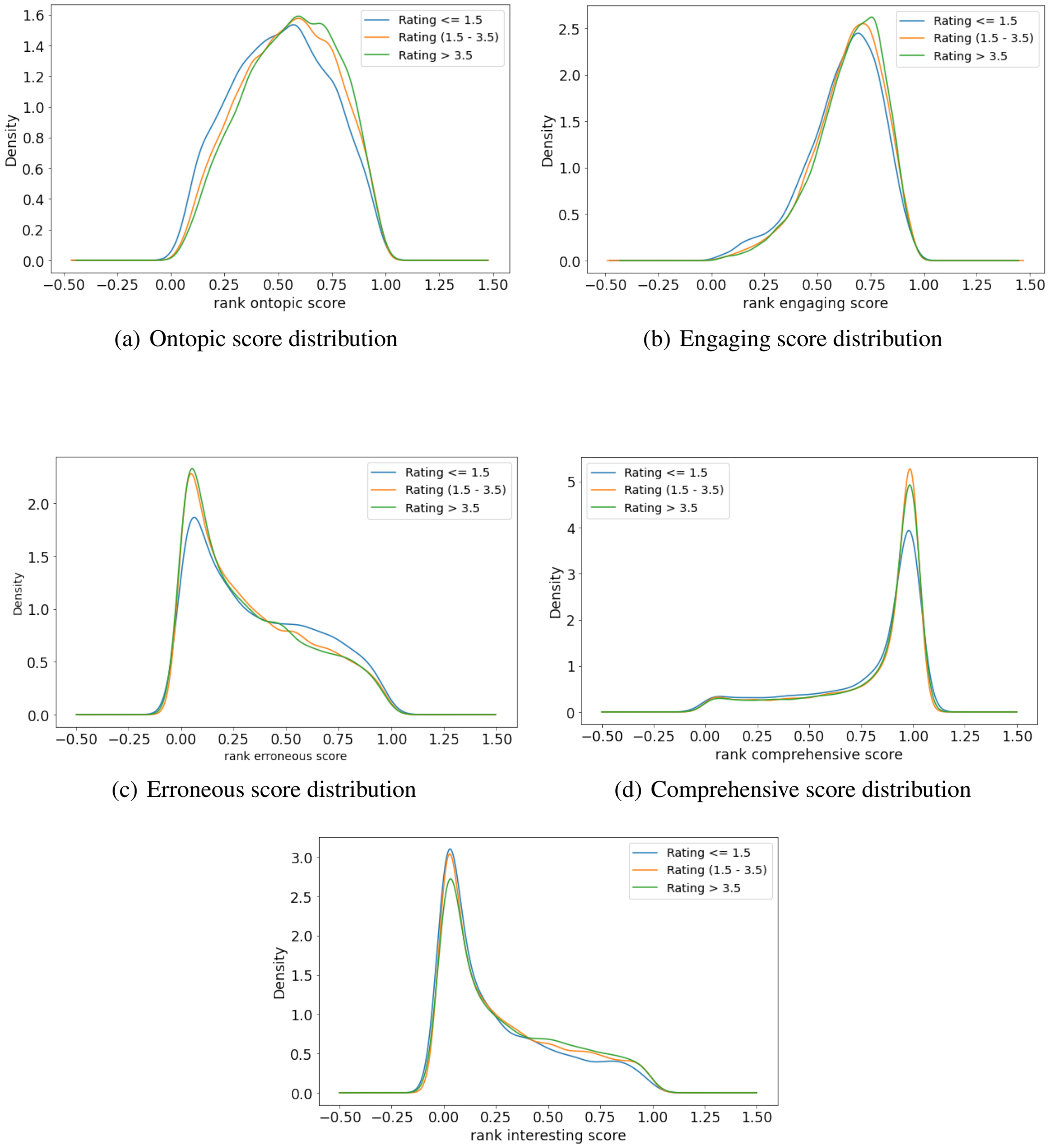
Table 10: Rating analysis per Generative Model.

As it can be seen in the Table 10, highest scores were given to the NRG, PRDRG and BlenderBot. The high value for LAUNCHER, which is not a generative model, reflects the high scores we get by using rule-based dialog management at some specific points of interaction as were described in section 2.5.4. Again, we can also observe the low ratings for the SENSITIVE control of inappropriate user behavior, so this is definitely an area still requiring more research and development. Also, DialogGPT and TransferTranfo present rating scores lower that the other generative models. An explanation for this is that they were among the latest models to be included in Genuine, so they still needed some more fine-tuning to make them suitable for the Alexa SocialBot scenario. In the case of TransferTranfo, it remained pending how to combine the interesting personalization capabilities of this powerful generative model with the richer content required by Alexa SocialBot interaction.

Besides, the performance of the retrieved knowledge incorporated into the PDNRG model requires additional analysis and fine-tuning. Finally, it is worth mentioning, that the rating score for TopicSwitch was found to be very relevant requiring further research on how to incorporate the dialog history, past topics, and mentioned entities.

3.3 Ratings analysis and Ranker

The third line of ratings analysis is directed to understand a key component of Genuine: the Ranker. As explained in section 2.4.3, the built-in ranker provides 5 different scores: ontopic, engaging, erroneous, interesting, comprehensive. Therefore, it is highly relevant to research on which one of these scores correlates better with the users' quality perception through rating. To this goal we monitored the statistical distributions of the 5 ranking scores for low, medium, and high rated dialogs. An example of this ratings monitorization is shown in Figure 5.



(a) Ontopic score distribution

(b) Engaging score distribution

(c) Erroneous score distribution

(d) Comprehensive score distribution

(e) Interesting score distribution

Figure 5: Ratings monitorization for low, medium and high scores in May.

It can be seen that ontopic and engaging scores are more related to users' ratings. However, the high overlap between distributions for different rating ranges show the difficulty in having objective metrics highly correlated with users' perception. Even our linear regression approach (sec. 2.4.3) may not be optimal. Therefore, in the next section we try to use an alternative option by evaluating the performance of a reference free metric to score our dialogues.

3.4 D-Score: Reference-free evaluation metric

Previously, in [Zhang et al., 2021a],[Banchs et al., 2015] and [D'Haro et al., 2019], we have addressed the problem of proposing reference-based dialogue system evaluation metrics for open-domain chatbots with high correlations with human evaluations. However, in [Zhang et al., 2021b], we recently introduced a free-reference metric for evaluating non-goal-oriented dialogue systems. This metric has been found to highly correlated with human evaluations on different datasets across multiple dimensions (e.g. Semantic Appropriateness, Logical Consistency, Avoiding Repetitions, Inquisitiveness, Interestingness, etc.) and for three challenging datasets: DSTC6 [Hori et al., 2019], DSTC7 [Galley et al., 2019], and Persona-Chat [See et al., 2019].

Through this metric we wanted to overcome some of the main limitations of the current metrics: a) they are laser-focused on a single aspect of the dialogue, b) lack of judgment of dialogue quality from a holistic perspective, c) specific to the evaluation task and hard to generalize, d) avoid using ground-truth references through the use of pre-training the model using unsupervised tasks, and e) to check their correlation with the optional human ratings provided to our chatbot at the end of the interaction. Briefly, the proposed metric is intended to be holistic since it is able to evaluate the following four common dimensions of a dialogue: a) Semantic appropriateness of responses, b) Context coherence, c) Language fluency, and d) Logical consistency. Figure 6 shows the original architecture of our proposed D-Score metric, as well as the four self-supervised tasks used to evaluate the four dimensions mentioned above.

The architecture is trained end-to-end in a multi-tasking manner. It takes into account the preceding and succeeding context to the current response we want to evaluate (for the competition, we did not use the succeeding context to be able to use the metric as an alternative for the pre-defined ranker in Cobot). First, it concatenates the preceding/succeeding and current response and passes it through the RoBERTa encoder (E) to generate a contextual representation for each word in the concatenated sentences (Ip and Is). Then, we take the activation of the second last transformer encoder layer of the encoder w.r.t. Ip and Is are extracted as their respective rich semantic vector representations. The second last layer is used to extract a more general contextualized representation, while the embeddings in the last layer are more tuned towards downstream tasks. Then, we encode the temporal flow at turn-level and token-level, that represents the coherence of a dialogue, using a bidirectional LSTM layer. Finally, we form a single vector representation by using an alignment layer.

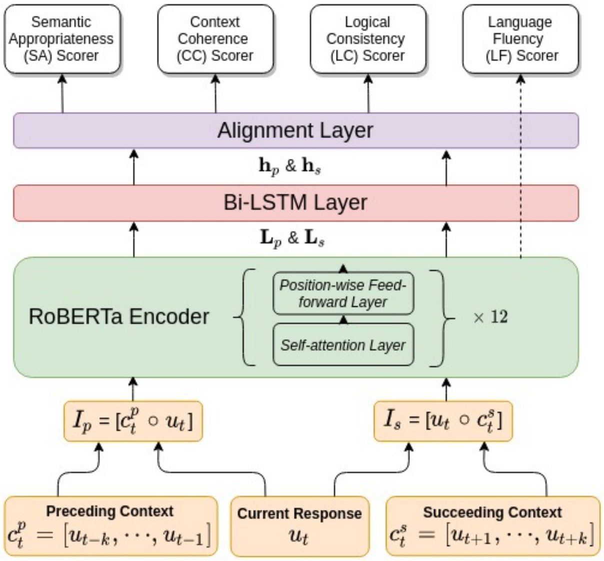

Semantic Appropriateness (SA) Scorer
Context Coherence (CC) Scorer
Logical Consistency (LC) Scorer
Language Fluency (LF) Scorer
Alignment Layer
$\mathbf{h}_p$ & $\mathbf{h}_s$
Bi-LSTM Layer
$\mathbf{L}_p$ & $\mathbf{L}_s$
RoBERTa Encoder
Position-wise Feed-forward Layer
Self-attention Layer
× 12
$I_p = [c_t^p \circ u_t]$
$I_s = [u_t \circ c_t^s]$
Preceding Context $c_t^p = [u_{t-k}, \cdots, u_{t-1}]$
Current Response u_t
Succeeding Context $c_t^s = [u_{t+1}, \cdots, u_{t+k}]$

Figure 6: D-Score architecture and proposed self-supervised tasks.

The four tasks in Fig. 6 can be considered as four independent scorers, each producing a quality indication for each linguistic dimension. The final composition of the four scores forms our D-score metric for the current utterance u_t. These four tasks are trained using a different self-response sampling policy per task.

For semantic appropriateness (SA), we adopted the approach of randomly selecting an utterance from another conversation to replace the original response. For the context coherence (CC), we use a swapping strategy where the current response is swapped with an utterance randomly sampled from either its preceding context or succeeding context.

For the logical consistency (LC), we cast this problem as a natural language inference (NLI) problem. To avoid the expensive manual creation of NLI labels, we follow the approach proposed in [Dziri et al., 2019]. First, entailment is found when the original current utterance occurs (i.e., human-human dialogues are generally consistent), if a random sentence from another dialogue is selected is considered contradiction. Besides, sentences in the context with word order shuffling changes the syntax and semantic information producing logical inconsistency. When the order of one sentence in the context is changed, we consider it as entailment if the shuffled sentence is close to the current one, but as neutral when it is from a far sentence to the current one.

Finally, for language fluency (LF) we employ sentence-level log-likelihood adapted from the perplexity score of a language model. Perplexity is defined as the average inverse probability of a sentence over the tokens, and a lower perplexity suggests higher fluency. To relate higher score value to higher fluency, we define a syntactic score as,

$$F_{LF}(u_t) = \frac{1}{y}\sum_{t}^{y} \log \mathcal{P}(w_t|w_{<t})$$

The response log-likelihood is an unbounded negative value. Hence, we apply min-max normalization to all the response scores in a test set to keep them between 0 and 1. The original metric is intended to be used for offline analysis since it uses the preceding and succeeding context. However, we made small adaptations to the original formulation to allow the metric to be used online by removing the requirement of the succeeding context.

In order to assess the results of our metric, we selected 990 rated dialogues with at least 5 turns interactions (a turn is defined here a one human and one chatbot interaction), resulting in a total number of 17465 turns. Our results using the same model reported in [Zhang et al., 2021b] are presented in table 11. The results show the Pearson and Spearman correlations between the human ratings and the scores produced by the D-score metric and the average scores produced by the Ranker. In order to evaluate the system at turn level, we simplified the problem by assigning the dialogue rating score provided by the human users to each chatbot response/turn in the dialogue. For the dialogue rating, we aggregate the individual scores obtained for each turn using D-score, then calculate the average score for the dialogue and compare against the human rating.

	Overall @ SGC4	
Dialogue Level	Pearson (p-value)	Spearman (p-value)
D-score	0.049 (0.02)	0.040 (0.05)
Avg. Ranker score	0.116 (0.2x10-3)	0.099 (0.002)
Turn Level	Pearson (p-value)	Spearman (p-value)
D-score	0.010 (0.05)	0.007 (0.18)
Avg. Ranker score	0.037 (0.9x10-6)	0.032 (3x10-5)

Table 11: System level and Turn level Pearson & Spearman correlation for 990 rated dialogues collected during the semi-finals round.

Unfortunately, these results proof the difficulties on using reference-free metrics on real data [Yeh et al., 2021]. Moreover, they also confirm our previous findings on how difficult is to predict ratings based on number of turns, length of the interactions, topic handled, etc. In the table, we can see that the ranker has some slightly better results due that our chatbot policy is based on the same ranker scores used here.

As future task, we consider fine-tuning the D-score model over our collected data for better comparison. Then, we want to assess the performance of the D-Score metric for response selection in contrast with the selection of the actual re-ranker. Finally, we want to check the performance of our latest designed model Dynaeval which is currently the state-of-the-art at dialogue level evaluation. This model generates a dynamic representation of the dialogue turns by considering different relationships between turns which is learnt using a graph convolutional network [Zhang et al., 2021c].

4 Conclusions and future work

In this paper, we have described in detail the architecture, modules, dialogue flow, and results analysis for our open domain Genuine^2 chatbot created for our first time participation in the Alexa Prize Socialbot Grand Challenge 4 (SGC4). Different to previous years, and thanks to the recent advancements in DNN generative systems, our chatbot is based on using multiple generators trained on different topics, intents, persona profiles, and usage of knowledge information. Given the large variability and quality of the responses generated by each generator, our dialogue manager main policy moves around two main dimensions: a) to maintain consistency and continuity between intents and topics with respect to the dialogue history, and b) rely on the assessment of the ranker module to select the candidate response based on selecting the best combination of multiple dimensions (engagement, comprehension, interestingness, topic coherence, and errorless).

Then, we also proposed traditional solutions based on pre-defined prompt answers to handle specific cases where we want to produce an even more reliable and coherent answer, but also to handle special cases (e.g., toxic users, switching topics, greeting new or previous users, jokes, sensible questions, etc.)

On the other hand, we also wanted our chatbot to create a more empathetic and enjoyable interaction by monitoring emotions in the user and chatbot prompts, avoiding repeated or non-coherent answers based on persona profiles, and even including some level of emotions in our chatbot responses.

Finally, we also presented an analysis of our chatbot ratings considering different dimensions: length of the interaction, topics we handle, and selected generator. Through this analysis, and by means of different tools we developed to analyze the interactions, we gave a glimpse on how challenging, but at the same time interesting, is the problem of creating enjoyable interactions.

Based on our invaluable experience in this challenge, we foresee as future work the development of metrics that could help to detect, offline and at real-time, those turns where our chatbot needs to be improved (e.g., due to lack of knowledge or hallucinations), situations where the user or the chatbot became stacked due to ASR errors or misunderstandings, inconsistencies/repetitiveness in the chatbot responses, but also creating more empathetical responses. Finally, we will work on improving the handling of sensitive topics, specially searching for solutions to avoid false positives and false negatives which could create frustration in the users or for toxic users to exploit weaknesses in the chatbot responses.

Acknowledgments

In first place, we want to thank Kate Bland, Savanna Stiff, Laureen Stubells, Shui Hu, and all the Amazon team that helped us to solve technical questions very quickly and easy. We also want to thank, from our university, to professors Luis A. Hernández and José M. Pardo for their support and experienced advise on speech technologies and conversational systems. Finally, we also thank our university management office for their support.

References

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

Ulrich Gnewuch, Stefan Morana, and Alexander Maedche. Towards designing cooperative and social conversational agents for customer service. In *ICIS*, 2017.

Ryan M Schuetzler, Mark Grimes, Justin Scott Giboney, and Joesph Buckman. Facilitating natural conversational agent interactions: lessons from a deception experiment. 2014.

Javier Cebrián, Ramón Martínez, Natalia Rodríguez, and Luis Fernando D'Haro. Considerations on creating conversational agents for multiple environments and users. *AI magazine special issue on Conversational AI (accepted, pending of publication)*, 2021.

Chandra Khatri, Behnam Hedayatnia, Anu Venkatesh, Jeff Nunn, Yi Pan, Qing Liu, Han Song, Anna Gottardi, Sanjeev Kwatra, Sanju Pancholi, Ming Cheng, Qinglang Chen, Lauren Stubel, Karthik Gopalakrishnan, Kate Bland, Raefer Gabriel, Arindam Mandal, Dilek Hakkani-Tur, Gene Hwang, Nate Michel, Eric King, and Rohit Prasad. Advancing the state of the art in open domain dialog systems through the alexa prize, 2018.

Hyojin Chin and Mun Yong Yi. Should an agent be ignoring it? a study of verbal abuse types and conversational agents' response styles. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.

Hyojin Chin, Lebogang Wame Molefi, and Mun Yong Yi. Empathy is all you need: How a conversational agent should respond to verbal abuse. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

Amanda Cercas Curry and Verena Rieser. A crowd-based evaluation of abuse response strategies in conversational agents. *arXiv preprint arXiv:1909.04387*, 2019.

Ashwin Paranjape, Abigail See, Kathleen Kenealy, Haojun Li, Amelia Hardy, Peng Qi, Kaushik Ram Sadagopan, Nguyet Minh Phu, Dilara Soylu, and Christopher D Manning. Neural generation meets real people: Towards emotionally engaging mixed-initiative conversations. *arXiv preprint arXiv:2008.12348*, 2020.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Agustín Manuel de los Riscos and Luis Fernando D'Haro. Toxicbot: A conversational agent to fight online hate speech. In *Conversational Dialogue Systems for the Next Decade*, pages 15–30. Springer, 2021.

Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. Challenges for toxic comment classification: An in-depth error analysis. *arXiv preprint arXiv:1809.07572*, 2018.

Jinchao Li, Baolin Peng, Sungjin Lee, Jianfeng Gao, Ryuichi Takanobu, Qi Zhu, Minlie Huang, Hannes Schulz, Adam Atkinson, and Mahmoud Adada. Results of the multi-domain task-completion dialog challenge. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, Eighth Dialog System Technology Challenge Workshop*, 2020.

Rafael E Banchs. Movie-dic: a movie dialogue corpus for research and development. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 203–207, 2012.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. *arXiv preprint arXiv:1106.3077*, 2011.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*, 2020.

Issa Annamoradnejad and Gohar Zoghi. Colbert: Using bert sentence embedding for humor detection. *arXiv preprint arXiv:2004.12765*, 2020.

Lawrence Philips. The double metaphone search algorithm. *C/C++ users journal*, 18(6):38–43, 2000.

Raefer Gabriel, Yang Liu, Anna Gottardi, Mihail Eric, Anju Khatri, Anjali Chadha, Qinlang Chen, Behnam Hedayatnia, Pankaj Rajan, Ali Binici, et al. Further advances in open domain dialog systems in the third alexa prize socialbot grand challenge. *Alexa Prize Proceedings*, 2020.

Kevin Clark and Christopher D Manning. Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667*, 2016.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, 2018.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*, 2017.

Sheng-Yeh Chen, Chao-Chun Hsu, Chuan-Chun Kuo, Lun-Wei Ku, et al. Emotionlines: An emotion corpus of multi-party conversations. *arXiv preprint arXiv:1802.08379*, 2018.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv preprint arXiv:1811.00207*, 2018.

Robert Plutchik. The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American scientist*, 89(4): 344–350, 2001.

Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*, 2019.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation, 2020.

Behnam Hedayatnia, Karthik Gopalakrishnan, Seokhwan Kim, Yang Liu, Mihail Eric, and Dilek Hakkani-Tur. Policy-driven neural response generation for knowledge-grounded dialogue systems. *arXiv preprint arXiv:2005.12529*, 2020.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*, 2019.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*, 2021.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

Chung Hoon Hong, Yuan Liang, Sagnik Sinha Roy, Arushi Jain, Vihang Agarwal, Ryan Draves, Zhizhuo Zhou, William Chen, Yujian Liu, Martha Miracky, et al. Audrey: A personalized open-domain conversational bot. *arXiv preprint arXiv:2011.05910*, 2020.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL `https://arxiv.org/abs/1908.10084`.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*, 2020.

Mario Rodríguez-Cantelar, Luis Fernando D'Haro, and Fernando Matía. Automatic evaluation of non-task oriented dialog systems by using sentence embeddings projections and their dynamics. In *Conversational Dialogue Systems for the Next Decade*, pages 71–84. Springer, 2021.

Chen Zhang, Luis Fernando D'Haro, Rafael E Banchs, Thomas Friedrichs, and Haizhou Li. Deep am-fm: Toolkit for automatic dialogue evaluation. In *Conversational Dialogue Systems for the Next Decade*, pages 53–69. Springer, 2021a.

Rafael E Banchs, Luis F D'Haro, and Haizhou Li. Adequacy–fluency metrics: Evaluating mt in the continuous space model framework. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):472–482, 2015.

Luis Fernando D'Haro, Rafael E Banchs, Chiori Hori, and Haizhou Li. Automatic evaluation of end-to-end dialog systems with adequacy-fluency metrics. *Computer Speech & Language*, 55: 200–215, 2019.

Chen Zhang, Grandee Lee, Luis Fernando D'Haro, and Haizhou Li. D-score: Holistic dialogue evaluation without reference. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021b.

Chiori Hori, Julien Perez, Ryuichiro Higashinaka, Takaaki Hori, Y-Lan Boureau, Michimasa Inaba, Yuiko Tsunomori, Tetsuro Takahashi, Koichiro Yoshino, and Seokhwan Kim. Overview of the sixth dialog system technology challenge: Dstc6. *Computer Speech & Language*, 55:1–25, 2019.

Michel Galley, Chris Brockett, Xiang Gao, Jianfeng Gao, and Bill Dolan. Grounded response generation task at dstc7. In *AAAI Dialog System Technology Challenges Workshop*, 2019.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. What makes a good conversation? how controllable attributes affect human judgments. *arXiv preprint arXiv:1902.08654*, 2019.

Nouha Dziri, Ehsan Kamalloo, Kory W Mathewson, and Osmar Zaiane. Evaluating coherence in dialogue systems using entailment. *arXiv preprint arXiv:1904.03371*, 2019.

Yi-Ting Yeh, Maxine Eskenazi, and Shikib Mehri. A comprehensive assessment of dialog evaluation metrics. *arXiv preprint arXiv:2106.03706*, 2021.

Chen Zhang, Yiming Chen, Luis Fernando D'Haro, Yan Zhang, Thomas Friedrichs, Grandee Lee, and Haizhou Li. Dynaeval: Unifying turn and dialogue level evaluation. *arXiv preprint arXiv:2106.01112*, 2021c.