Task Oriented Dialogue Systems

Sunit Singh
Ohio State University
Columbus, USA
singh.1790@osu.edu

Abstract

In this report we explore task oriented dialogue systems and present "Tacobot", a dialogue system built to assist users with Cooking and DIY tasks. Tacobot follows a user-centered principle and is equipped with elaborate language understanding, dialogue management and response generation modules to engage the user in a diverse and immersive conversation aimed at task execution. We discuss data augmentation strategies to craft a suitable training resource for specialist models used in Tacobot. We also discuss their training methodologies and deployment frameworks. Tacobot is developed to be an Alexa Skill and is completely deployed on the AWS cloud infrastructure allowing it to be highly scalable and available. We also open-source out in-house generated large scale Cooking QA dataset to contribute to the community.

1 Introduction

We introduce Tacobot, a conversational assistant designed to aid users in executing complex cooking and home improvement tasks. Tacobot aims to deliver a user-focused and interactive experience for users to actively engage with, during the execution of a task. In order to achieve this, Tacobot has been designed to excel in natural language understanding, offer flexibility in dialogue management and generate informative and engaging responses.

To tackle the domain specific nature of the task categories, Tacobot's training corpus incorporated heavily from content generated by GPT4 and human-in-the-loop approaches to annotate and evaluate the system. By translating user behavior into practical design principles and developing a comprehensive test and feedback suite, Tacobot demonstrated commendable performance in the Amazon Alexa Taskbot Challenge, achieving an average user satisfaction rating of 3.55 out of 5 and a task completion rate of nearly 35 %.

Figure 1: Dialogue showing first few turns

2 Challenges

There are several critical challenges that need to be addressed in the space of task oriented dialogue. Firstly, there is a tendency to optimize functional objectives, often at the cost of user's overall experience. Secondly, while there exist a plethora of cooking datasets, bridging the gap between merely assisting users with task instructions and establishing task related commonsense, requires nuanced data collection strategies and human annotations to align with humans executing the task. For instance, during a cooking task, questions requiring details of food by description of color, texture, ingredient substitute for a particular flavor are fairly common. These require the dialogue system to have cooking related physical and temporal commonsense knowledge in order to best answer qualitative questions as accurately as possible. On the other hand, the system should be accurate with the quantitative measures involved with cooking or DIY tasks. Finally, striking a balance between task execution and organic interaction through task related or unrelated question answering and chit-chat, poses

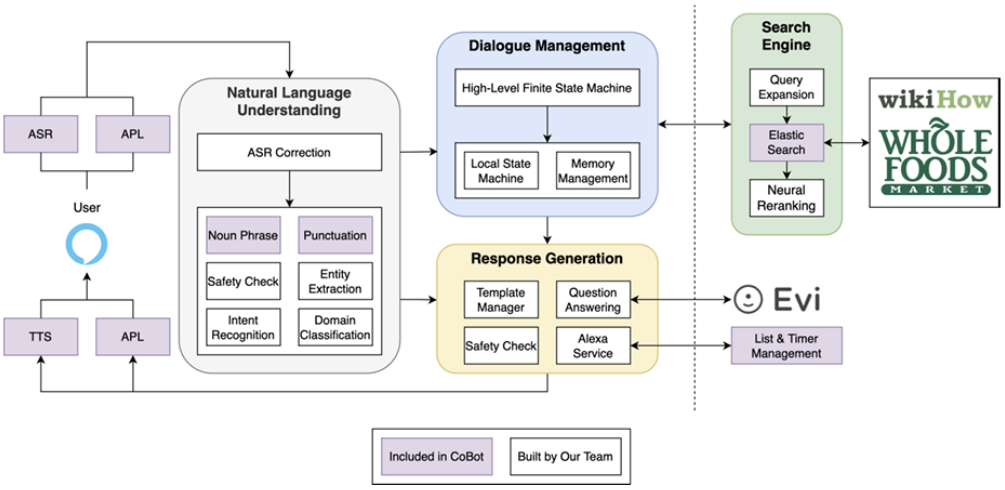

Figure 2: Tacobot Overall System Design

challenges in the dialogue progression strategy.

3 System Overview

Tacobot is built on the CoBot framework (Khatri et al., 2018). The dialogue agent is deployed as an Amazon Skill with which the user interacts by speaking or via a touch screen device(Amazon Echo Show). Alexa's Automatic Spech Recognition (ASR) transcribes the user's speech into a text utterance and Alexa Presentation Language (APL) handles the argument-value pairs captured from touch events.

The user utterance is first processed by the Natural Language Understanding (NLU) module to extract useful information from the utterance. This stage involves ASR error correction and entity and intent extraction through models running on remote AWS EC2 instances. The Cobot framework runs the instances in parallel which results in reduced latency in running the NLU phase.

The results of the NLU phase are handled by the Dialogue Management (DM) module which acts as an orchestrator to maintain the state and context of the dialogue and advance the conversation towards task completion by selecting the response generation module. The dialogue state and context is managed through a low latency NoSQL DynamoDb persistent storage.

The response generation phase comprises of several rule-based and text generation models running concurrently on Amazon Lambda and EC2 instances, respectively. For the rule-based models we use Amazon Lambda since the response framework is essentially event-driven and rule based or template driven models don't have complexity in terms of processing or bootstrapping upon event triggers. The text response is rendered by Alexa's SSML Text-To-Speech (TTS) or APL services before being relayed to the user.

3.1 Natural Language Understanding

The NLU phase is the first phase that executes on each dialogue turn. It comprises of pre-trained language model and rule-based approaches to build four components:

1. **ASR Error Correction.** This does a lookup against a set of common ASR errors in certain dialogue states, and also uses a neural approach that performs error correction based on the utterance context.

2. **Hierarchichal Intent Recognition.** This module annotates one of 11 dialogue intents, from 4 broad categories, for the user utterance. Following is the categorical intents breakdown :

   (a) **Sentiment :** Affirm, Negate, Neutral

   (b) **Commands :** Navigational commands like Forward(X steps), Backward(X Steps), Go To Step X, More / Less Choice, Request Detail, Complete Task

   (c) **Utilities :** Request, Help, Question, Timer

   (d) **Exception :** Out-of-Domain Input, Incomplete Utterance

The intent annotation is carried out using a multi-label classification model trained on conversational intent detection data and data augmented with GPT-3 prompting. This collected data is further augmented by replacing slot values with placeholders to generate synthetic utterances, which are filled with various sampled values depending on the task, to generate training examples for the intent classifier. Additionally, BERT-based models fine-tuned on span detection dataset to domain classification tasks are used in identifying the task name from among the task domains ie. Cooking or DIY tasks.

3.2 Dialogue Management

The dialogue management is essentially a hierarchichal finite state machine which comprises of three phases :

1. **Task Search.** Given the task domain and task name from the NLU pipeline's result, the Task Search module issues an Amazon OpenSearch query against the indexed task catalogue to retrieve relevant tasks from which the user can choose. Elastic search overemphasizes lexical similarity between input query and task titles leading to semantic mismatches in search results on occasions. To mitigate this issue, query expansion is applied to user queries, which generally tend to be short, by adding related words, including lemmatized verb, nouns to improve recall in OpenSearch results. The task search queries WholeFoodsMarket data for cooking tasks and WikiHow data for DIY tasks, and a neural re-ranking module is used on the OpenSearch results to further improve the task recommendations. This stage also presents the user with clarification questions regarding cuisine, diet constraints in order to retrieve tasks that closely match the user's preferences.

2. **Task Preparation.** This phase begins after the user has selected a task. In this phase the user gets to review detailed information about the task before they choose to begin task execution, or alternatively go back to the task search stage.

3. **Task Excecution.** After the user chooses to begin the task, the dialogue flow transitions into the task execution phase where Tacobot guides the user in a step-by-step manner to complete the task. This phase accommodates navigational intents in order to skip steps, review steps. This is also an interactive phase where the user can ask questions related to the task or outside the task context. A comprehensive question-answering module is responsible for handling user queries in the best possible manner, with the aim of increasing user engagement with the dialogue agent.

4. **Utility.** This module allows Tacobot users to access various features like Alexa list and timer management without affecting the dialogue state.

3.3 Response Generation

The response generation module comprises of 4 major components, namely, Template-Based Generation, Chit-Chat, "People Also Ask" and a Question Answering module. These modules use the dialogue state information and conversation context to generate responses to be presented to the user in the dialogue turn.

1. **Template-Based Generation.** In the Template-Based response generation, several response templates are curated by native speakers and several pre-written paraphrases are generated. At response composition time, a template is randomly selected to generate diverse, human-like responses. The sentence-level templates contain placeholders or "slots" which are populated with data values using composition rules defined by the states of the hierarchical finite-state machine during the Dialogue Management phase. Tacobot uses regular expressions to segment unstructured texts into conversation-friendly phrases which are combined based on the underlying sentence template.

2. **Chit Chat.** During a dialogue with a task bot, conversations generally extend across multiple turns and users often desire casual talk, unrelated to or tangential to the task at hand. The chit-chat module enables users to have open domain conversations from diverse topics and is essential in raising user engagement. It uses joint intent and entity tracking to focus on topics relevant to the user's current utterance. The recognized entities are searched over web

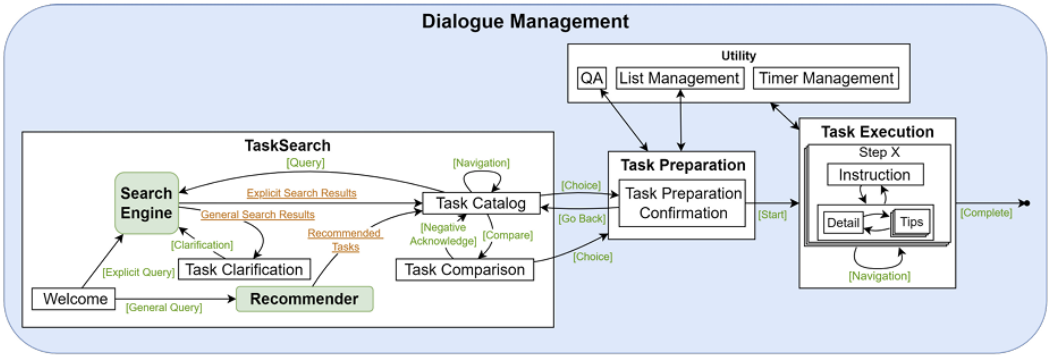

Figure 3: Dialogue Management Module

sources (Wikipedia and Google) through web crawls to extract information around those topics. The chit-chat response generation incorporates various strategies like neural chat, categories like food, sports, travel and cultures, aliens etc. and BlenderBot-3B to generate open domain responses. The chit-chat module is also responsible for reverting the conversation to the ongoing task after a few turns of task unrelated dialogue.

3. **People Also Ask.** "People Also Ask" is another module aimed at enriching the user-bot interaction by suggesting engaging task-related conversation topics to the user. Tacobot leverages Google's People Also Ask (PAK) feature to provide a list of related questions and summarized answers from web pages. PAK data is collected by extracting 30k common keywords from task titles from the WholeFoodsMarket and Wikihow corpuses, resulting in 494k QA pairs. During task execution, the PAK content is displayed to the users as additional information about the current task step. The user can interact with PAK prompts by asking follow-up questions to the information which triggers the chit-chat module or the question answering module.

4 Question Answering

Tacobot's Question Answering (QA) module is an essential module throughout the task related dialogue. Tacobot's QA supports a wide variety of questions from the user, related to the task or on an unrelated topic. While the agent's main objective is the successful completion of the task, it is also designed to entertain user's questions about the task. Tacobot uses an instruction tuned model

# of samples	context	answerable QA	unanswerable QA
	1832	4431	752
avg len (# of words)	context	question	answer
	93.9	9.76	17.9

Figure 4: Annotated data statistics for WikiHow QA dataset

on an augmented cooking recipe and DIY tasks dataset to answer task related questions. Tacobot can also answer open book questions about any unrelated topic, for which it uses pretrained open sourced instruction tuned models. The QA module is a group of neural models specializing in a particular question type, operating together in a priority based order.

4.1 Knowledge Base and Datasets

Tacobot uses a blend of retrieval-based and generative approaches which require task-related data to be carefully curated to task-specific needs.

1. **Task Knowledge Bases** - For retrieval based models Tacobot uses the WholeFoodsApi for cooking tasks and WikiHow Api for DIY tasks. Additionally, frequently asked questions (FAQs) are used from Wikihow articles to form indexed FAQ pairs. The retrieval is based on cosine similarity between the user question and the FAQ question index.

2. **Human Data Annotation** - We observe that MRC models trained on span detection datasets like SQuAD (Rajpurkar et al., 2016) fail to domain specific robustness for task related questions. To remedy this issue, we sample 1832 paragraphs as context from task related WikiHow articles and a group of 15 data annotators create upto 3 question-answer pairs

for each paragraph. Keeping with the annotation scheme of SQuAD2.0 (Rajpurkar et al., 2018), the answers to the questions are either sentences in the context for answerable questions or a special '[No Answer]' token for unanswerable questions. In total, 5183 QA pairs including 752 *unanswerable* questions are created.

3. **LLM generated QA dataset** - In order to train a generative model for better understanding and reasoning capability about the task, we prompt GPT-4 with task instructions from the RecipeQa (Yagcioglu et al., 2018) and Amazon Wizard of Tasks (Choi et al., 2022) datasets, and design a prompt to generate upto 20 questions related to the each recipe. RecipeQa consists of 20,000 unique instructional recipes with multiple modalities like titles, descriptions and aligned set of images. Amazon Wizard of Tasks dataset contains 549 cooking and DIY task conversations with upto 18,000 dialogue turns. Each conversation turn contains metadata about the step such as "Step Relevance", "Usefulness".

   We design a prompt that aims to generate QA pairs from an open-sourced LLM (in our case, GPT-3.5), that exhibit deeper understanding about the task. We create a small catalogue of task-related reasoning questions (60 QA pairs across 10 cooking tasks, and 30 questions across 6 DIY tasks) using a human-in-the-loop approach. The recipes for the catalogue are picked by us and we manually inspect the generated QA and choose the most informative ones. This catalogue serves as a few-shot prompt for the LLM to generate similar QA pairs for all the tasks in our composite dataset. During the LLM prompting step we randomly sample a task and its related QA pairs from the curated catalogue and use that as a few shot template in our prompt. The dataset contains over roughly 30,000 tasks comprising of task instructions and question andwer pairs related to the task. In total, the dataset has about 920,000 QA pairs with 31 QA pairs per task on average. The questions are broadly of 4 types :

   (a) Physical Commonsense - Ingredient/tool substitutes, visual/texture description

   (b) Temporal Commonsense - Time required for a particular step in the instructions(duration), altering sequence of steps(ordering), how often to execute a step(frequency)

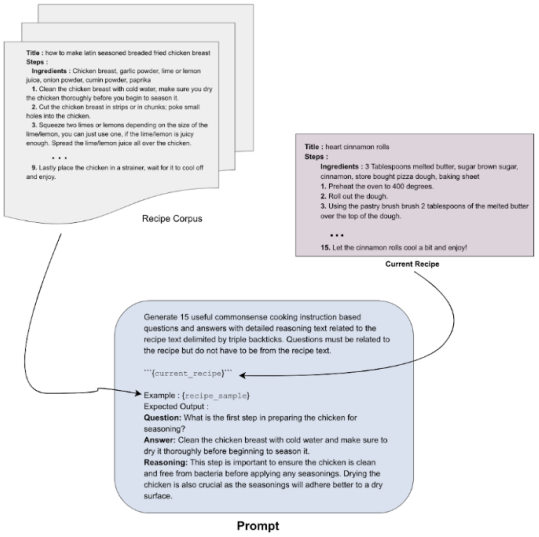

Figure 5: Data augmentation prompt template

   We open source the LLM generated dataset to the community as an additional large scale QA resource for question answering tasks.

4.2 Task-related QA

1. **Context-dependent QA** We first use a "Domain Type" classifier model which classifies the user's question into 3 types (Cooking, DIY, None). We identify whether the question is task related or open-ended in the beginning because these two types are handled by different response generation(RG) flows.

   Once the domain of the question is identified as Cooking or DIY, a question type classifier s used to classify the user question into 5 types (MRC, FAQ, Factual, Ingredient, Substitute) and 3 types under a DIY task (MRC, FAQ, Factual). The question types are handled by individual rule-based or neural models trained to retrieve information from the context of the current task or an external task corpus. The retrieval is predicated on high precision matches between the question and the task corpora, so as to either fetch highly relevant answers or fail over to the next priority stage in the QA workflow.

2. **Context-Independent QA** Though the context dependent RG modules have a rich knowl-

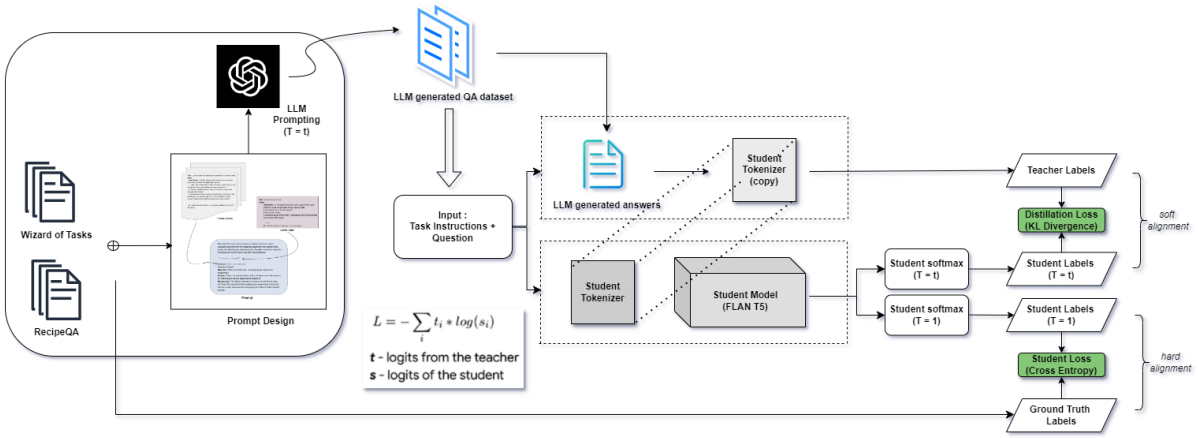

Figure 6: Knowledge Distillation from LLM prompted QA

edge resource to handle a wide variety of questions related to the user's task, it can not handle questions which need extraneous information about the nature of the task, or reasoning involved in the task. For instance, *"How much longer should I leave the chicken in the oven if I want it a little toasty"* required the agent to reason about the dish. We observed such questions are frequently asked by the user executing a task, and it is infeasible to cover the vast possibilities of user's questions. We therefore instruction tune generative models specialized for cooking and DIY tasks, separately. Keeping in mind, the low latency requirements, we choose to train a 3B parameter FLAN XL model using the LLM augmented dataset(refer LLM generated QA as a knowledge distillation resource.

Grounding the context-independent RG's responses in the brief and objective responses from the augmentation-seeding datasets(Amazon Wizard of Tasks and RecipeQa), and using the detailed LLM generated responses as distillation source allows our model to add useful detail while answering the user's question.

4.2.1 Training

The question answering data is converted to an instruction tuning format, wherein the prompt comprises of the prompting text, task instructions followed by the user's question. The FLAN T5 XL is a 3B parameter model and we use low rank adaptation(LoRA) from the Transformers PEFT (Mangrulkar et al., 2022) library.

```
lora_config = LoraConfig(
    r=32, # Rank
    lora_alpha=32,
    target_modules=["q", "v"],
    lora_dropout=0.05,
    bias="none",
    task_type=TaskType.SEQ_2_SEQ_LM # FLAN-T5
)
```

LoRA Parameterization Results	
Total model parameters	2868631552
Trainable model parameters	18874368
percentage of trainable model parameters	0.66%

Figure 7: Low Rank Addaptation of the Student Model

The model is jointly trained on 2 objectives :

(a) **Student CE Loss.** The student loss is a cross-entropy loss using the original Wizard of Tasks and RecipeQa dataset answers as ground truth and the model's predictions. This loss induces an alignment between the model's responses and the ground truth answers.

$$CELoss = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \quad (1)$$

(b) **Distillation KD Loss.** The teacher loss is a Kullback-Leibler(KL) divergence measure between the teacher logits that are produced in proxy using the student

model's tokenizer (T5Tokenizer) and student (model's) logits, softened by a Temperature factor $T = t$.

$$KL(\hat{y}||y) = \sum_{c=1}^{M} \hat{y}_{c,T=t} \log \frac{\hat{y}_{c,T=t}}{y_{c,T=t}}$$

4.2.2 Evaluation

To evaluate the distilled model serving as our context independent RG, we carry out a two-stage approach. In the first stage we use an entailment score to measure whether the long-form answers generated by the model entail the ground truth answers. We observe a high entailment score of 0.91 using the distilled model.

$$ModelResponse \models GroundTruth$$

We also evaluate the BertScore (Zhang* et al., 2020) between the model's generated response and the ground truth answers. BertScore uses cosine similarity to correlate candidate and reference sentences, thereby taking sentence semantics into account. We observe a BertScore f1 value of 0.54 for sentence-level evaluations. We ascribe the drop in this score due to the disparity in the content of model generated and ground truth answers.

In the second stage, we measure the model's online performance as part of the holistic system. For this we use A/B testing against the legacy system and gauge user engagement with the dialogue. We measure user engagement via two implicit metrics, "turn count" (the total number of dialogue turns between user and the agent) and "task completion rate" (ratio of the total dialogues that lead to successful completion of task), and an explicit metric in the form of customer rating for the conversation. We observe improvements across all fronts with the new model, with a 37 % increment in turn count, 26 % increase in task success rate. In the customer rating, we see a small improvement from an average rating of 3.42 to 3.68; we observe that the user rating suffers drastically from lack of relevant tasks presented to the user which causes the user to leave a poor rating without entering into dialogue with the agent.

4.3 Open Domain QA

Although Tacobot is a task oriented dialogue system, we provide open domain question answering capability in order to provide an engaging experience to the user. Open Domain QA refers to questions unrelated to the current task or, unrelated to Cooking/DIY tasks altogether. These are essentially outside of the agent's purview. Open source generalist LLMs are a good choice to respond to such questions. As this agent is in the format of an Amazon Alexa Skill, it can not use third party APIs for inference, in keeping with customer data privacy regulations. We evaluate 3 models as candidates for open domain QA :

(a) **Amazon Alexa Teacher Model.** This is a 20B parameter sequence-to-sequence model trained on a mixture of Common Crawl (mC4) and Wikipedia data across 12 languages using denoising and Causal Language Modeling (CLM) tasks (Saleh Soltan, 2022).

(b) **Openassistant Pythia Model.** This is a variant of EleutherAi's Pythia (Biderman et al., 2023) family of LLMs, fine-tuned on Open Assistant Conversations dataset, viz. a crowd-sourced human generated and human annotated assistant-style conversations.

(c) **Amazon EVI.** This is a closed source, open domain response retrieval system provided by Amazon.

We test the 3 models across 3 datasets to evaluate their generalization capability. We use syntactical measures like Rouge score (Lin, 2004) and phrase embedding similarity scores to score the model responses against the evaluation dataset's ground truth responses. To measure semantic similarity we use the BleuRT (Sellam et al., 2020) score, a regression model metric trained on ratings data, suitable for conveying textual similarities in natural language generation data.

(a) **ComplexWebQuestions Dataset.** ComplexWebQuestions (Talmor and Berant, 2018) is a large dataset containing 34,689 QA pairs. The questions require reasoning over multiple web snippets, and the answers to the questions can be

Model	ComplexWebQuestions			Commonsense QA			Curated Tasks QA		
	Textual Similarity Score	Rouge L Score (F1)	BleuRT	Textual Similarity Score	Rouge L Score (F1)	BleuRT	Textual Similarity Score	Rouge L Score (F1)	BleuRT
Alexa TM 20B	0.37	0.25	**0.33**	**0.52**	**0.21**	**0.34**	0.41	0.27	0.40
OpenAssistant Pythia	**0.44**	**0.27**	0.29	0.48	0.19	**0.34**	**0.67**	**0.39**	**0.48**
Amazon EVI	0.21	0.08	0.11	0.16	0.04	0.09	0.19	0.12	0.17

Figure 8: Evaluation results of out-of-domain QA models.

found via search engine interactions. It also contains comprehension task and a semantic parsing task datasets, but we only use the factoid QA pairs for our evaluation.

(b) **Commonsense QA Dataset.** Commonsense QA (Talmor et al., 2019) consists of 12,247 commonsense questions with multiple-choice answers, generated by Amazon Mechanical Turk workers using "concept-chaining" with the help of ConceptNet (Speer et al., 2018)

(c) **In-house Task Dataset.** Finally we also test the open domain model's performance on task related questions by evaluating against our annotated dataset Human Annotated QA Dataset.

We observe that the Alexa TM 20B and OpenAssistant Pythia models show comparable performance on the ComplexWebQuestions and Commonsense QA datasets. For task related questions, the OpenAssistant Pythia model performs comparatively better. One issue with the responses generated from OOD models is potential harmful content such as offensive speech, harmful task suggestions, financial or medical advice. To mitigate these issues we use an Offensive Speech Classifier API provided by Amazon to filter such content out. We also design rules to detect model hallucinations in generated responses and ether clean up repetetive phrases in the model responses or filter out the response altogether. Amazon EVI responder does not issue a response for more than 55 % of the questions on average, resulting in low overall scores. But, it does does not produce harmful text nor does it hallucinate. Thus, we use Amazon EVI as a fall-back responder in the out-of-domain QA pipeline.

5 Deploying the System

Tacobot is built using the Cobot framework developed by Amazon. The end-to-end architecture of the system is hosted on the AWS cloud in a scalable and highly available manner, designed to serve low latency responses to the user. In this section we architect a AWS solution to deploy the system keeping in mind the aforementioned features. We try to decouple the system modules as far as possible to prevent single-point-of-failure(SPOF) in the complete workflow, i.e. natural language understanding(NLU) to dialogue management(DM) and finally response generation(RG). While the execution of a single conversation turn is stateful in nature, other aspects of the system such as dialogue state persistence is managed in an asynchronous manner.

The application web content is cached in CDNs like Amazon Cloudfront. The dialogue orchestration is responsible for integrating the different modules, NLU, Dialogue Manager, Response Generation in the dialogue workglow. The orchestrator is deployed on a Amazon Lambda function. Rule-based NLU annotators and Response Generators eg. Noun phrase Extractor, Ingredients and Substitutes also use Amazon Lambda. Amazon Lambda is a serverless computing service that allows deployments without managing servers. With the help of Lambda Handler, a function stub, Lambda handles everything required to run and scale the code with high availability. This makes Lambda ideal for lightweight applications, as it supports short-term, event-driven processes efficiently. The compute intensive prediction models are deployed on Amazon EC2 instances. Communication with the different modules if via REST Api.

6 Closing Notes and Future Work

In this paper, we introduced Tacobot, a modular task-oriented dialogue system that assists users with a wide catalogue of daily tasks. We proposed a comprehensive set of modules to work collaboratively in order to execute task-related dialogue and to offer versatility in the interaction to maintain user engagement. We used various data augmentation techniques and leveraged LLMs to allow Tacobot to have a deep understanding of the task. We open-source the framework and datasets, providing valuable resources in the space of task related user-agent interactions.

As part of further work on Tacobot, we aim to expand the tasks catalogue and include more task types. Tacobot would also benefit with the use of multimodal approaches. This would enable the bot to have visual understanding of the items used in task execution. We also aim to develop generalist models to handle various types of interactions in place of specialist models for the different QA classes. Generalist models would benefit from a holistic knowledge about various tasks, and thereby facilitate greater transfer of knowledge across tasks.

References

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling.

Jason Choi, Saar Kuzi, Nikhita Vedula, Jie Zhao, Giuseppe Castellucci, Marcus Collins, Shervin Malmasi, Oleg Rokhlenko, and Eugene Agichtein. 2022. Wizard of tasks: A novel conversational dataset for solving real-world tasks in conversational settings. In *COLING 2022*.

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. 2018. Advancing the state of the art in open domain dialog systems through the alexa prize.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Jack FitzGerald Rahul Gupta Wael Hamza Haidar Khan Charith Peris Stephen Rawls Andy Rosenbaum Anna Rumshisky Chandana Satya Prakash Mukund Sridhar Fabian Triefenbach Apurv Verma Gokhan Tur Prem Natarajan Saleh Soltan, Shankar Ananthakrishnan. 2022. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2018. Conceptnet 5.5: An open multilingual graph of general knowledge.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1368, Brussels, Belgium. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.