TACO 2.0: A Task-Oriented Dialogue System with Mixed Initiatives and Multi-Modal Interaction

Lingbo Mo[*], **Huanli Gong**[†], **Sunit Singh**[†], **Chang-You Tai**[†]
Tianhao Zang[†], **Tianshu Zhang**[†], **Huan Sun**[‡]

The Ohio State University

{mo.169, gong.545, singh.1790, tai.97,
zang.107, zhang.11535, sun.397}@osu.edu

Abstract

In the inaugural Alexa Prize TaskBot Challenge, we introduced TACO 1.0, a task-oriented digital assistant, designed to guide users through multi-step tasks in cooking and home improvement. Equipped with a suite of components including language understanding, dialogue management, and response generation, bolstered by a search engine, TACO 1.0 set a robust foundation in user-centered assistance. Building on this, we present **TACO 2.0**, aspiring to deliver a more *collaborative* and *engaging* dialogue experience. Towards this end, we refine our mechanisms to better accommodate the dynamic nature of real-world conversations, supporting mixed initiatives from both users and agents. In terms of user initiative, we develop an upgraded hierarchical intent recognition module and a more powerful question-answering system to accurately comprehend and respond to user needs. To cater to agent initiative, we incorporate a chit-chat functionality, allowing for multi-turn casual conversations. Furthermore, we delve into a series of strategies for multi-modal interaction to continuously improve user engagement.

1 Introduction

Recent developments in Task-Oriented Dialogue (TOD) systems have demonstrated promising performance on accomplishing user goals in a conversational manner (Mehri et al., 2019; Semantic Machines et al., 2020; Zhang et al., 2020; Peng et al., 2021; Su et al., 2022; Mo et al., 2022). Traditional TOD systems are often user-driven, where users provide information and direct the system to perform a task, such as reserving a hotel or booking flight tickets. Different from the setting like that, we developed TACO 1.0 (Chen et al., 2022), a task-oriented dialogue system built for the first Alexa Prize TaskBot Challenge, which assists users in completing multi-step cooking and home improvement (DIY) tasks. It harnesses a robust suite of capabilities, including language understanding, dialogue management, and response generation, fortified by a search engine. Expanding on this foundation, we introduce TACO 2.0 in the second Alexa Prize TaskBot Challenge (Agichtein et al., 2023), an advanced system intended to accommodate mixed initiatives via a collaborative process. More than mere task completion, TACO 2.0 places strong emphasis on multi-modal interactivity and user engagement, thereby offering a holistic, user-centered conversational experience.

However, several challenges arise as we strive to achieve our vision: (1) Given the inherent ambiguity and variability of natural language, accurately recognizing user intent is a formidable challenge.

[*]Team lead.
[†]Team members in alphabetical order.
[‡]Faculty advisor.

2nd Proceedings of Alexa Prize TaskBot (Alexa Prize 2023).

(2) Users may pose a broad spectrum of questions, and the hurdles to overcome include accurately comprehending user inquiries, accessing a wide array of knowledge sources, and executing advanced reasoning and inference to deliver comprehensive responses. (3) The digital agent cannot merely serve as a passive information provider within a dialogue system. Strategically incorporating mixed initiatives is critical for an engaging and dynamic conversational experience. (4) Beyond providing textual step-by-step instructions, the effective coordination of visual and verbal modalities is vital to enhance user engagement and enrich the overall dialogue experience.

To this end, TACO 1.0 explored data augmentation strategies by utilizing large language models (LLMs) to construct a robust intent recognition module. Meanwhile, it introduced a question answering system primarily focused on task-related questions. Building on this groundwork, we highlight several advancements in TACO 2.0 including: (1) we develop a more nuanced, hierarchical intent recognition module, designed to cater to diverse user initiatives. (2) To address various queries from the user, we build a more comprehensive question answering system that spans both task-related and out-of-domain questions. This system expands to integrates a variety of knowledge sources, including in-domain database and LLMs. (3) We incorporate a chit-chat functionality that enables users to discuss other topics of interest and supports mixed initiatives from both the user and the digital agent. (4) To enhance user engagement, we've delved into multiple strategies for multi-modal interaction. These include presenting pertinent visuals through multi-modal retrieval and alignment, generating visual fun facts via text-to-image synthesis, and introducing a series of Graphical User Interface (GUI) designs to ensure a smooth and user-friendly dialogue experience.

2 System Overview

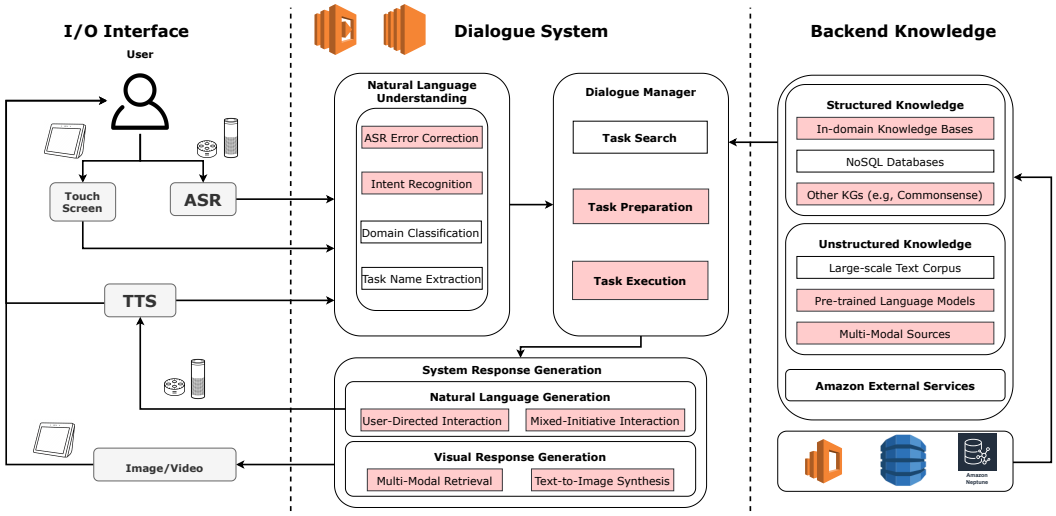

Figure 1: Overview of TACO 2.0 system architecture. We highlight the enhanced components in TACO 2.0 by boxing them with a red-colored background.

TACO 2.0 is constructed using the CoBot framework (Khatri et al., 2018), enabling a user to engage with our system either through voice commands or by using the touchscreen interface, if available. For voice interaction, Alexa's Automatic Speech Recognition (ASR) technology transcribes the user's spoken language into text. Conversely, in the touchscreen mode, touch events are characterized as a collection of argument-value pairs using Alexa Presentation Language (APL).

Figure 1 offers a detailed illustration of the process upon user input reception. The Natural Language Understanding (NLU) module initiates the process by preprocessing the user's utterance to identify their intent. Subsequently, the Dialogue Management (DM) module, designed with a hierarchical finite state machine, manages the dialogue flow, resolves exceptions, and advances the conversation towards task completion.

At each conversational turn, the Response Generation (RG) module crafts the utterance, drawing upon the required knowledge and other modalities for an enriched user interaction. The robust support of

our well-organized knowledge backend and search engine underpins each component. This allows for connection with diverse sources to better aid the user. The finalized textual or multimodal response, rendered by Alexa's SSML Text-To-Speech (TTS) and APL services, is then conveyed to the user, ensuring a collaborative and engaging experience.

3 Natural Language Understanding

The NLU pipeline runs at the beginning of each dialogue turn. We employ a fusion of potent pre-trained language models and dependable rule-based approaches, consisting of four key components: (1) ASR Error Correction, (2) Intent Recognition, (3) Task Name Extraction, and (4) Task Domain Classification. We continue to employ methods from TACO 1.0 for task name extraction and task domain classification. Essentially, task name extraction is formulated as a span extraction task and fine-tuned with a BERT-based model (Devlin et al., 2018). Task domain classification, on the other hand, utilizes a separate BERT-based binary classifier to distinguish between cooking and DIY tasks, thereby allowing the bot to offer tailored dialogue experiences. In TACO 2.0, we emphasize our effort to refine ASR error correction and intent recognition, as detailed in the subsequent sections.

3.1 ASR Error Correction

ASR errors can hinder the bot from understanding user intentions, degrading the overall user experience. To address this, TACO 1.0 annotated common ASR errors in specific dialogue states and corrected them using strict string matching. In TACO 2.0, we developed a more advanced algorithm to further tackle ASR errors. We construct three knowledge bases for this purpose, encompassing common tasks, recipes, and commands derived from real user conversations. Entries in these bases include typical tasks and food names used by users, as well as common commands under each state of the bot. We also utilize data from sources like WikiHow for additional task and food names. To expedite the process of sound matching, we generate phonemes for all the entries in our knowledge base. Statistics pertaining to the knowledge base are showcased in Table 1.

Category	Size	Example
Task	4399	"make a mini greenhouse": M EY1 K AH0 M IH1 N IY0 G R IY1 N HH AW2 S
Food	2126	"blueberry smoothie": B L UW1 B EH2 R IY0 S M UW1 DH IY0
Command	694	"Welcome": { "what's your favorite": W AH1 T S Y AO1 R F EY1 V ER0 IH0 T, ...} "TaskPreparation": { "let's cancel": L EH1 T S K AE1 N S AH0 L, ... } "TaskExecution": { "next step": N EH1 K S T S T EH1 P, ... }

Table 1: Statistics about the knowledge base (until July 26).

The aforementioned knowledge bases underpin our pipeline designed to correct ASR errors. We utilize Amazon's speech recognition tool, retaining high-confidence sentences, deleting low-confidence ones, and triggering error correction for those with moderate confidence (0.4 - 0.9). We choose knowledge base according to whether the sentence contains potential "Task" or "Food" slots based on ASR results, and utilize a grapheme-to-phoneme tool[4] to calculate pronunciation similarity with knowledge base utterances. Tokens will be replaced by the most similarly pronounced knowledge base utterance if they are similar enough. Sentence confidence is then recalculated based on the original sentence and replacement tokens. We compare remaining sentences, selecting the one with the highest confidence. Figure 2 illustrates the detailed flow chart. We did explore the use of language models for correcting ASR errors but found our knowledge base approach more fitting due to reasons elucidated in Section 7.1.

3.2 Intent Recognition

In order to accommodate a wide array of user initiatives, we build a hierarchical intent recognition module that organizes multiple intents into four categories, as detailed in Table 2. Furthermore, real-world user initiatives often encompass several intents within one single utterance. For example, *"No, I want to know how to paint the wall."* should be classified as both **Sentiment (**`Negate`**)** and **Task**

[4]https://github.com/Kyubyong/g2p

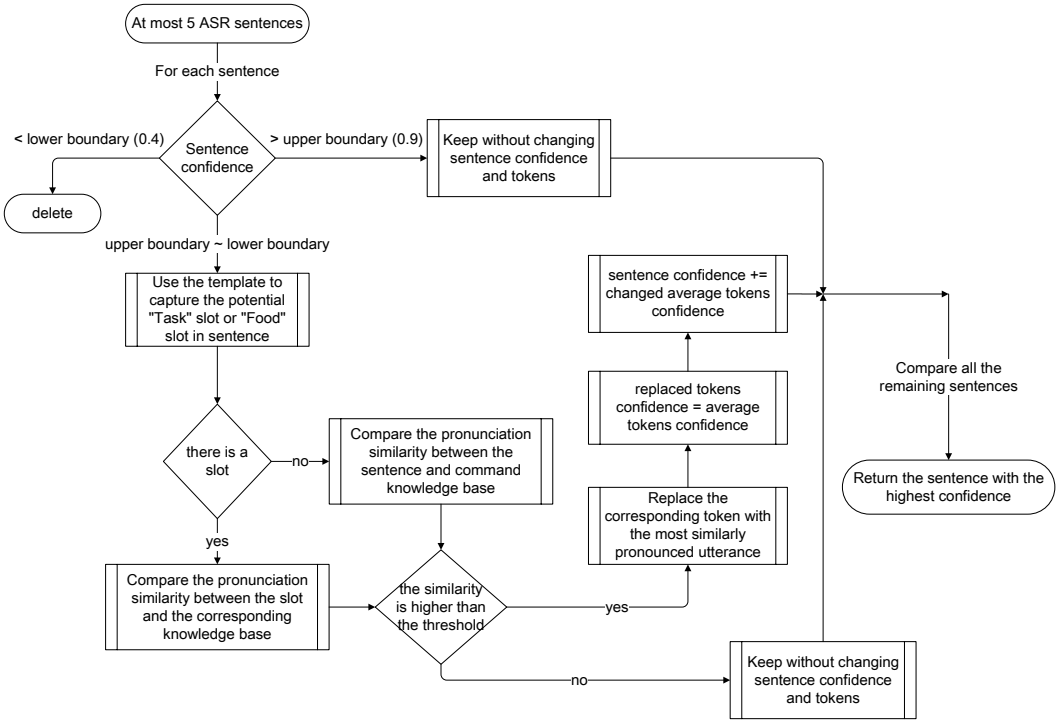

Figure 2: ASR error correction flow chart.

Request intents. Accordingly, we address intent recognition as a multi-label classification problem and filter model predictions according to the dialogue state.

To develop a high-quality multi-label classification model despite limited data, we employ data augmentation and domain adaptation techniques. We leverage existing datasets (Rastogi et al., 2019) for common intents like **Sentiment** and **Question**, while utilizing the in-context learning capability of GPT-3 (Brown et al., 2020) for other intents. By synthesizing initial utterances with intent descriptions and few-shot examples, we create a foundation for training data. To expand the dataset, we transform synthetic utterances into templates, substituting slot values with placeholders and filling them with sampled values to generate actual training utterances. Additionally, we incorporate linguistic rules, neural paraphrase models, and user noise, such as filler words, to enhance data diversity and improve the robustness of our intent recognition module.

Category	**Description**
Sentiment	The user can confirm or reject the bot's response on each turn, leading to three labels: `Affirm`, `Negate`, and `Neutral`, indicating the user utterance's polarity.
Commands	The user can drive the conversation using these commands: **Task Request**, **Navigation** (to view candidate tasks or walk through the steps), **Detail Request**, **PAK Request**, **Task Complete**, and **Stop** to terminate the conversation at any time.
Utilities	We use a **Question** intent to capture user questions and a **Chat** intent for casual talk.
Exception	To avoid unintentional changes in dialogue states, we have one additional intent for out-of-domain inputs, such as incomplete utterances and greetings.

Table 2: Categories of detailed intents to support diverse user initiatives.

Task Request vs. QA During the task search phase, users frequently use queries to find specific tasks. However, these questions may serve purposes other than task requests, potentially resulting in intent classification challenges. For instance, consider these examples: "What is the best smoothie?" versus "How do you say my love in French?" While the first query indicates a task request intent, the latter simply suggests a question from the user. To better differentiate these intentions, we've leveraged Distilbert (Sanh et al., 2019) to develop a specialized classifier that determines whether a user's query constitutes a task request. To train this model, we gathered real-world conversational

data and manually annotated a thousand question samples. Check Section 7.3 for more training details and results.

4 Dialogue Management

We design a hierarchical finite state machine for the DM component, consisting of three phases: Task Search, Task Preparation, and Task Execution. Each phase comprises multiple fine-grained dialogue states, as depicted in Figure 3.

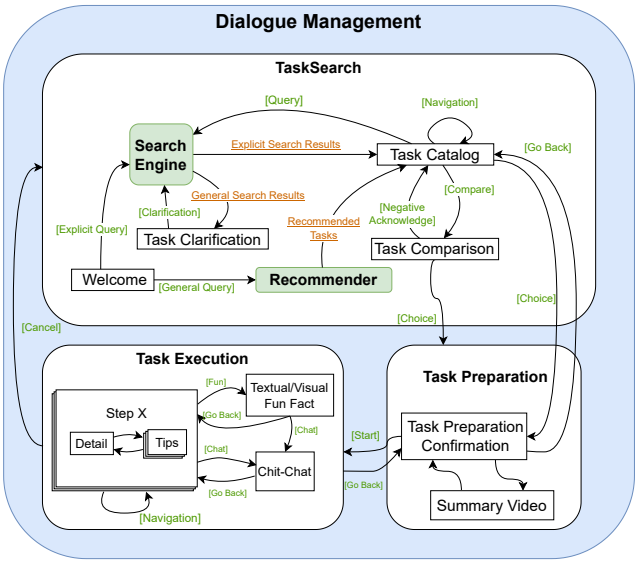

Figure 3: Dialogue management diagram. White boxes represent dialogue states and green boxes represent supporting modules. Bidirectional edges represent reflexive transitions. Green texts represent user intent and orange texts denote search engine output.

Task Search Phase. Task Search serves as the initial phase in TACO 2.0, allowing users to explore DIY tasks or recipes. Users have the flexibility to issue specific queries and receive search results from our backend search engine if they have a particular idea in mind. Alternatively, they can request the bot to recommend interesting tasks. In both cases, our bot presents and compares candidate tasks retrieved by the search engine to assist users in making informed choices. Moreover, TACO 2.0 actively clarifies dietary and cuisine preferences for recipe searches, providing more accurate results. After making a choice, users proceed to the Task Preparation phase.

Task Preparation Phase. Users are eligible to transition into the Task Preparation phase exclusively after selecting a candidate task. During this phase, users review comprehensive information about the chosen task and decide whether to proceed. To offer an overview of the task, we provide a *summarizing video* that visually encapsulates what the task entails. If users change their mind, they can return to Task Search to find an alternative task. Conversely, they can commit to the task, leading them to enter the Task Execution phase.

Task Execution Phase. In this phase, TACO 2.0 guides the user through task instructions step by step, aided by helpful tools like the QA module. Each task step has its own unique state. For detailed tasks, we split long steps into manageable instructions, details, and tips, ensuring easy comprehension. We also incorporate a *chit-chat feature* for casual conversations on user-interested topics, along with relevant fun facts and *creative text-to-image synthesized visuals*. Users have the flexibility to cancel the ongoing task and revert to the Task Search phase to start anew.

Once a user input is received, DM transitions states and selects suitable response generators. Each conversation turn within a phase is assigned a hierarchical dialogue state, offering flexible transitions at various levels. We also keep a history stack of dialogue states, letting users easily revert to prior

states. Invalid user intents, which don't trigger state transitions, instead activate context-specific help, guiding users in their dialogue progression. This design offers users a stable and adaptable conversation experience.

5 Response Generator

Our response generation module capitalizes on a hybrid methodology, combining both template-based and neural-based techniques. In the template-based method, we employ handcrafted conditional rules to fill slots with retrieved data and assemble templated segments. These response templates, stored as phrase and sentence segments, were meticulously curated by native speakers and included various pre-written paraphrases in the first challenge. During composition, they are randomly selected to produce diverse and human-like responses. We arrange these templates and their corresponding composition rules based on the high-level states in our hierarchical finite-state machine. For the neural-based approaches, our primary focus is on the Question-Answering (QA) component, a vital feature of task-oriented dialogue systems. We will discuss this in detail in the subsequent section.

5.1 Question Answering

Throughout the conversation, users may pose a range of questions that generally fall into two categories: *task-related* or *out-of-domain*. In TACO 1.0, our promary focus was on developing approaches for task-related questions, fortified by a question type classifier. For completeness, we describe them in Section 5.1.1. In TACO 2.0, we continue to utilize this classifier and further design a more comprehensive QA system to better manage both categories of questions. On one hand, it bolsters the ability to answer task-related questions by fine-tuning the FLAN-T5 model (Wei et al.). On the other hand, it broadens the scope to include out-of-domain QA, incorporating a Pythia model (Biderman et al., 2023) trained on OpenAssistant conversations (Köpf et al., 2023), supplemented by the open-domain EVI system and the Alexa Teacher Model (ATM) (FitzGerald et al., 2022) provided by Amazon, to ensure a more holistic response generation.

5.1.1 Question Type Classifier

To fulfill user needs more accurately, we further categorize task-related questions into four nuanced types. These include in-context Machine Reading Comprehension (MRC) for context-dependent QA, a Frequently-Asked Questions (FAQ) retrieval module for DIY tasks, and a rule-based ingredient and substitute QA module for cooking tasks. Coupled with Out-Of-Domain (OOD) questions, we employ a question type classifier that sorts user queries into five categories (`MRC, FAQ, OOD, Ingredient, Substitute`) within a cooking task and three categories within a DIY task (`MRC, FAQ, OOD`). To more effectively distinguish between these question types, we combine the instructions for the current step (if available) with the input question, feeding this concatenated sequence into a `Roberta-base` classifier. For each question type, we sampled 5,000 instances for training purposes, with examples detailed in Table 3.

Question Type	Example	Context
MRC	Use what tool to blend?	add a 14-ounce can of sweetened condensed milk, unsweetened natural cocoa powder... use a whisk to blend the ingredients until they're completely mixed, and set the bowl aside. it's normal for the mixture to become very thick as you mix it.
OOD	What is the capital of US?	
FAQ	How should I know the ingredients are completely mixed?	
Substitute	I don't have condensed milk, can I use something else?	
Ingredient	How much cocoa powder do I need?	

Table 3: Examples of different types of questions.

5.1.2 Task-Related QA

For the MRC, FAQ, Ingredient, and Substitute QA modules, we continue to use the methods established in TACO 1.0. In the **MRC** module, we annotate an in-context QA dataset and refine an extractive QA model using Roberta-base (Liu et al., 2019). For the **FAQ** module, we gather QA pairs from the Community QA section of WikiHow articles and employ a retrieval module based

on the cosine similarity between question embeddings. In the **Ingredient QA** module, we use a high-recall string matching mechanism to extract ingredients mentioned by users from the recipe ingredient list. Lastly, in the **Substitute QA** module, we assemble a substitution dataset covering 200 frequently used ingredients to provide substitution suggestions. For more details, please refer to our prior technical report (Chen et al., 2022).

FLAN-T5. We further fine-tune a FLAN-T5 Large model (Wei et al.) for task-related questions. This serves as a strong backup, stepping in when the four primary QA modules cannot provide an answer. To facilitate this, we prompt ChatGPT (GPT-3.5) and GPT-4, using recipes and DIY tasks from the Amazon Wizard-of-Tasks dataset (Choi et al., 2022), along with recipes from the RecipeQA dataset (Yagcioglu et al., 2018) to generate task-related commonsense question-answer pairs. We apply one-shot prompting by citing an example context and its corresponding expected output. The prompt is constructed by sampling a random recipe from our recipe corpus as the prompt example and the "current recipe" for which we aim to generate QA pairs. Figure 4 depicts our prompt construction for generating recipe-specific question-answer pairs. This dataset comprises of about 920,000 QA pairs across roughly 30,000 recipes and 170 DIY tasks. It serves as a comprehensive resource for fine-tuning the decoder blocks of the FLAN-T5 model.

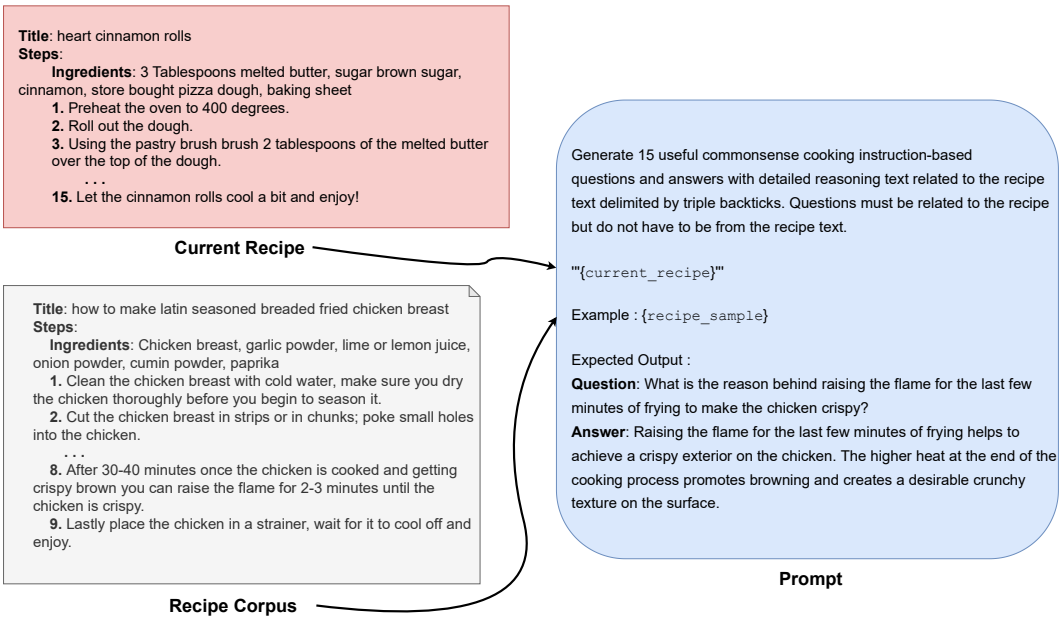

Figure 4: Creating a one-shot learning based prompt for ChatGPT.

5.1.3 Out-Of-Domain QA

EVI, ATM and Pythia. We direct out-of-domain questions to the EVI module. We also use the conversationally trained 20B variant of the Alexa Teacher Model(ATM) (FitzGerald et al., 2022) suite and Pythia (Biderman et al., 2023) 6.9B parameter model trained by OpenAssistant for out-of-domain questions. These comprise of questions unrelated to the task. We observe the Pythia model performs better on factoid QA, while the ATM handles non-factoid QA better. To optimize the models' performance, we configure the context based on the dialogue state. For instance, during the task execution stage, we provide a longer context (3 previous turns) as users are more likely to ask questions related to the current task step. The model demonstrates robustness in handling context switches, which occur when the user is in the middle of a task and asks a question unrelated to it. The conversational tone of the ATM's generated responses makes it particularly suitable for task-unrelated questions, adding to its effectiveness in such scenarios.

6 User Engagement

6.1 Chit-Chat

In the real-world dialogues with users, instead of only focusing on the task, they tend to show an interest in having a casual talk with the bot from time to time. To furnish a superior user experience, we plan to provide chit-chat functionality in TACO 2.0, enabling users to engage in versatile and engaging conversations. We adopt template-based strategy to identify the user intent for entering and exiting chit-chat. Taking inspiration from the social chatbot, Chirpy Cardinal (Chi et al., 2022), our chit-chat module consists of three components: *Entity Tracker*, *Chit-Chat Response Generator*, and *Intent Identification Model*.

Entity Tracker. It serves to monitor the entities throughout the chit-chat. This makes the generated responses better align with user intentions and exhibit specificity with regards to the current topic. Based on the recognized entities, TACO 2.0 can then access web sources (Wikipedia and Google), and allow the user to explore intriguing information centering around these entities.

Chit-Chat Response Generator. This component refers to Chirpy and incorporates various response generators there, providing for flexible and varied conversations. In particular, we integrate Neural Chat, Categories, Food, Aliens, Wiki, and Transition response generators to address broad topics of interest. Neural Chat distills a single model from BlenderBot-3B (Roller et al., 2021) with 9 decoder layers to generate open-domain responses. The Categories and Food generators are designed to elicit entity-related responses through the use of templates. Transition, on the other hand, facilitates the shift between entities and responds to diverse user inputs. Utilizing a hybrid template-based approach, Wiki enables users to discover engaging information about entities in a conversational style. Lastly, Aliens is a five-part monologue series about the topic of extraterrestrial existence.

Intent Identification Model. It endeavors to determine whether the user wishes to continue a particular topic or shift to a new one. It is noted that TACO 2.0 will proactively prompt the user to return to the ongoing task after a few turns of chit-chat. How to naturally transition between chit-chat and task-oriented turns requires prolonged efforts.

6.2 Multi-Modal Interaction

In TACO 2.0, we've explored a series of strategies for multi-modal interaction, with the goal of creating a more interactive and engaging dialogue experience. These strategies span all three phases throughout the conversation: task search, task preparation, and task execution.

6.2.1 Task Search

Personalized Search Buttons. In TACO 1.0, we introduced a single-turn clarification for recipe searches. With TACO 2.0, we aim to enhance the multi-modal experience by introducing interactive constraint buttons on the screen. These buttons allow users to flexibly specify their preferences. For recipes, these buttons could cover areas such as diet constraints, cuisines, and courses, while for DIY tasks, they could relate to ratings and views. Initially, search results would be presented without incorporating any constraints. However, as a user selects a specific constraint by clicking a button, the displayed recipes or tasks will dynamically update to reflect the selected constraint.

6.2.2 Task Preparation

Summary Video. To enhance user satisfaction and engagement, we've developed a summary video feature as part of our multi-modal functionality. This new, fully customized interface includes features like a video player window, progress drag bar, play button, and options to change videos. Users can access this interface by clicking the video button on the preparation screen. This opens a relevant video to assist users in accomplishing tasks, such as following a recipe or completing a WikiHow task. For detailed instructions, users can manipulate the progress bar or use the play button to pause the video at any time. Furthermore, users can easily return to the preparation interface by using the back button, allowing for quick reference to recipe ingredients or other information as needed.

6.2.3 Task Execution

Dynamic Effect. In order to increase the vividness and give users better experience during the interaction with our bot, we apply Ken Burns effect (Niklaus et al., 2019) on each still image of each recipe step to create a zoom in and zoom out video clip. More specifically, the method consists of a depth prediction model which predicts scene depth from the input image and a context-aware depth-based view synthesis model to generate the video results.

Visual Fun. We plan to infuse the visual fun elements into our bot. We leverage ChatGPT[5] to generate a collection of intriguing prompts and then use DALL·E[6] to generate high-quality images given the prompts. Those generated images demonstrate fictional scenarios where each element is real but their combination is unusual. We prompt users to see those funny images during the conversation if the users are interested. The key entity in each image binds with the key entity in the recipe name. As Figure 5 shows, when the user looks for a recipe for a white cake and wants to see some interesting pictures, our bot will show an interesting image which is related to "cake".

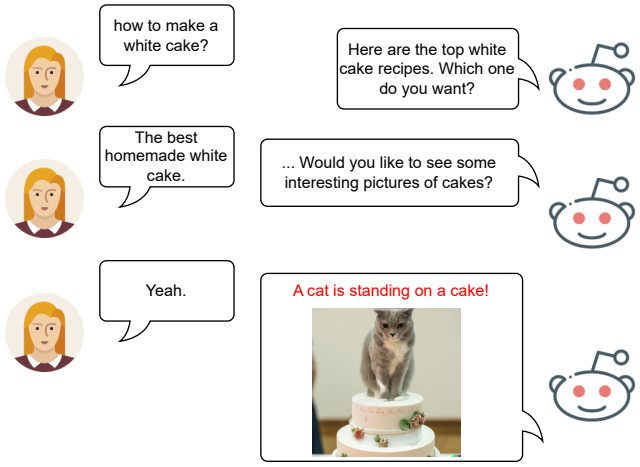

Figure 5: Visual fun during the interaction between the user and the bot .

Flip Card Design. On the execution page, we plan to add a special flip card design to spice interaction up. To be more concrete, we have a card contained with a useful question related to the execution step. When users touch the card, it will flip to the other side which shows the answer to this question. We hope by such a design, users can not only get the answer but also have fun during the exploring process as the interaction is presented in a way that reveals the mystery.

7 Evaluation and Analysis

7.1 ASR Error Correction

To evaluate different correction methods, we've created a large scale self-supervised dataset by comprehensively simulating noise on the speech data of the WikiHow sentences. First, we apply Speech Synthesis Markup Language (SSML) tags to the sentences from WikiHow to change the prosody of the resulting speech. To produce comprehensive speech, volume, rate, and pitch are randomly selected for each speech. Second, we use a text-to-speech (TTS) tool[7] to convert the tagged text into speech. Third, we add a random background noise[8] to the speech. To simulate the situation where the user does not immediately turn off the microphone after speaking, we add a silence that lasts up to 3 seconds after the speech. Finally, we feed the noisy speech to an ASR tool to generate transcriptions and keep them with Word Error Rate (WER) in a certain range. Following the

[5] `https://chat.openai.com`
[6] `https://openai.com/research/dall-e`
[7] The TTS tool we can use is Amazon Polly (`https://aws.amazon.com/polly/`)
[8] We use background noise from `https://www.pacdv.com/sounds/ambience_sounds.html`

aforementioned steps, we collect a total of 400,914 samples based on task titles and user questions on WikiHow website, and split them into training set (80%), test set (10%) and development set (10%).

We utilized this dataset to pre-train various language models, including general Seq2Seq models such as BART (Lewis et al., 2020) and T5 (Raffel et al., 2020), as well as an ASR error correction model, specifically ConstDecoder (Yang et al., 2022). ConstDecoder yielded the best performance, which recovers 23% of sentences in the test set. However, by leveraging additional information including state and confidence scores, the knowledge base method (mentioned in Section 3.1) can achieve up to 50% accuracy and benefit from shorter latency. More importantly, this method ensures that the output either originates from the original sentence or from the knowledge base, unlike model-based methods that can potentially generate content unlikely to be uttered by the user in certain states. As a result, we opt for the knowledge base method for ASR error correction.

7.2 Question Answering

Task-Related QA. We tested the question type classifier and machine reading comprehension model with manually annotated test sets in the first challenge. For completeness, we include the results here. For the question type classier, we evaluated it with 500 questions and observed an overall accuracy of 94%, which indicates the classifier is proficient at deciding the question types.

For the machine reading comprehension (extractive QA) model, we evaluate it on two test sets: (1) our own annotated test set and (2) questions from real conversations with TACO 1.0 (12/10/21 - 02/08/22), with answers annotated by a team member. As shown in Table 4, UnifiedQA and Roberta achieve comparable performance on *answerable* questions. However, UnifiedQA performs much worse in identifying unanswerable questions on both test sets. This is reasonable as UnifiedQA was pretrained on data sources which do not contain many unanswerable questions. To provide better user experience, it is more advisable to give no answer than some random answer. Thus, we adopt Roberta, an extractive model, as the backbone of our QA module.

	Test set (team created)		Test set (from real user)	
	Answerable (436)	Unanswerable (84)	Answerable	Unanswerable
UnifiedQA	39.3	5.9	10.3	22.4
+finetuning	**69.9**	49.2	72.6	43.2
Roberta	40.1	48.8	40.3	88.3
+finetuning	69.7	**69.1**	**72.3**	**88.9**

Table 4: Accuracy (Exact Match) of QA models. Number in parenthesis means test set size.

Out-Of-Domain QA. During the current challenge, we assess the performance of the Pythia 6.9B model and both variants of the Alexa Teacher Model — the vanilla ATM and the conversationally trained ATM — for out-of-domain questions. We employ the ComQA (Abujabal et al., 2019) and Commonsense QA (Talmor et al., 2019) datasets for this evaluation. ComQA, sourced from the community question-answering website, WikiAnswers, consists of 11,214 factoid questions that cannot be addressed by commercial search engines, requiring the model to leverage its inherent commonsense reasoning capabilities for responses. Meanwhile, CommonsenseQA is a dataset of 12,102 multiple-choice questions, curated across different facets of commonsense knowledge to predict accurate answers.

We use three metrics for the QA evaluation, measuring the performance of the generated answers from token-level semantics to sentence-level semantics.

- **Rouge Score** (Lin, 2004) - Measures token-level (Rouge-1 and Rouge-2) and phrase-level (Rouge-L) similarities.
- **Sentence Similarity** - Measures semantic similarity at sentence level by computing cosine similarity between sentence embeddings produced by a sentence transformer model trained on sentence similarity task.
- **BleuRT** (Sellam et al., 2020) - A transfer learning-based metric for natural language generation. Bleurt is a trained metric, a regression model trained on ratings data. The model is based on Bert

and RemBert. It captures non-trivial semantic similarities between sentences instead of just token level similarities.

Meanwhile, response generation latency is another critical metric we monitor. To gauge this, we experiment with various parameters such as context-window size, maximum token limit of generated response, text pre-processing and post-processing techniques. While text processing approaches do introduce a latency overhead, they are instrumental in generating high-quality responses. As illustrated in Table 5, Pythia achieves the best performance on question answering metrics, while the conversationally trained ATM shows the advantage for its shorter latency. Consequently, we integrate these two models into our QA system to answer out-of-domain questions.

Model	Rouge F1	Sentence Similarity	BleuRT	Avg. Latency (sec)
ATM (Vanilla)	0.09	0.41	0.32	6.5
ATM (Conv.)	0.14	0.46	0.40	**1.7**
Pythia	**0.28**	**0.67**	**0.48**	4.5

Table 5: QA evaluation results of Pythia and Alexa Teacher Model variants.

7.3 Intent Recognition: Task Request vs. QA

Model	Precision	Recall	F1
DistilBERT	0.94	0.91	0.92

Table 6: Intent identification results to differentiate Task Request and QA.

We've trained Distilbert Sanh et al. (2019) as a classifier to further distinguish task requests and question intents during the task search phase. To train this model, utterances from real user interactions were hand-labeled, categorized as either "task request intent" or "non-task request intent". We configured the model's hyper-parameters with a learning rate set to 0.00001, a batch size of 16, and a dropout rate of 0.1, leaving other parameters at their default settings. We evaluated the model's effectiveness in identifying user intents with a test set comprising 200 manually annotated samples. As demonstrated in Table 6, the trained model achieved a noteworthy overall F1 score of 91%. This high accuracy indicates the proficiency of the proposed classifier.

8 Conclusion and Future Work

In conclusion, we present TACO 2.0, a multi-modal task-oriented dialogue system designed to assist users with complex tasks in their daily life. We outline a series of modules with corresponding strategies, all aimed at providing a collaborative and engaging task bot experience. TACO 2.0 has demonstrated to be competitive most of the time in the semifinals, and our evaluative analysis further validates the effectiveness of our current deployed designs. Looking ahead, we plan to incorporate several enriching features, including chit-chat functionality, visual fun via text-to-image synthesis, a flip card design, and other improved GUI designs, with the goal of enhancing the user dialogue experience even further.

Acknowledgments

We would like to thank Amazon for financial and technical support as well as colleagues in the OSU NLP group for their valuable comments and feedback.

References

Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. 2019. ComQA: A community-sourced dataset for complex factoid question answering with paraphrase clusters. 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 307–317, Minneapolis, Minnesota. Association for Computational Linguistics.

Eugene Agichtein, Michael Johnston, Anna Gottardi, Cris Flagg, Lavina Vaz, Hangjie Shi, Desheng Zhang, Leslie Ball, Shaohua Liu, Luke Dai, Daniel Pressel, Prasoon Goyal, Lucy Hu, Osman Ipek, Sattvik Sahai, Yao Lu, Yang Liu, Dilek Hakkani-Tür, Shui Hu, Heather Rocker, James Jeun, Akshaya Iyengar, Arindam Mandal, Saar Kuzi, Nikhita Vedula, Oleg Rokhlenko, Giuseppe Castellucci, Jason Ingyu Choi, Kate Bland, , Yoelle Maarek, and Reza Ghanadan. 2023. Alexa, let's work together: Introducing the second alexa prize taskbot challenge. In *Alexa Prize TaskBot Challenge 2 Proceedings*.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Shijie Chen, Ziru Chen, Xiang Deng, Ashley Lewis, Lingbo Mo, Samuel Stevens, Zhen Wang, Xiang Yue, Tianshu Zhang, Yu Su, et al. 2022. Bootstrapping a user-centered task-oriented dialogue system. *arXiv preprint arXiv:2207.05223*.

Ethan A. Chi, Ashwin Paranjape, Abigail See, Caleb Chiam, Trenton Chang, Kathleen Kenealy, Swee Kiat Lim, Amelia Hardy, Chetanya Rastogi, Haojun Li, Alexander Iyabor, Yutong He, Hari Sowrirajan, Peng Qi, Kaushik Ram Sadagopan, Nguyet Minh Phu, Dilara Soylu, Jillian Tang, Avanika Narayan, Giovanni Campagna, and Christopher Manning. 2022. Neural generation meets real people: Building a social, informative open-domain dialogue agent. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 376–395, Edinburgh, UK. Association for Computational Linguistics.

Jason Ingyu 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 *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3514–3529, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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

Jack FitzGerald, Shankar Ananthakrishnan, Konstantine Arkoudas, Davide Bernardi, Abhishek Bhagia, Claudio Delli Bovi, Jin Cao, Rakesh Chada, Amit Chauhan, Luoxin Chen, Anurag Dwarakanath, Satyam Dwivedi, Turan Gojayev, Karthik Gopalakrishnan, Thomas Gueudre, Dilek Hakkani-Tur, Wael Hamza, Jonathan J. Hüser, Kevin Martin Jose, Haidar Khan, Beiye Liu, Jianhua Lu, Alessandro Manzotti, Pradeep Natarajan, Karolina Owczarzak, Gokmen Oz, Enrico Palumbo, Charith Peris, Chandana Satya Prakash, Stephen Rawls, Andy Rosenbaum, Anjali Shenoy, Saleh Soltan, Mukund Harakere Sridhar, Lizhen Tan, Fabian Triefenbach, Pan Wei, Haiyang Yu, Shuai Zheng, Gokhan Tur, and Prem Natarajan. 2022. Alexa teacher model. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM.

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-Tür, 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. *CoRR*, abs/1812.10757.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations – democratizing large language model alignment.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence

pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7871–7880.

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.

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

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. In *20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 165.

Lingbo Mo, Ashley Lewis, Huan Sun, and Michael White. 2022. Towards transparent interactive semantic parsing via step-by-step correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 322–342.

Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 2019. 3d ken burns effect from a single image. *ACM Transactions on Graphics*, 38(6):184:1–184:15.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:907–824.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *CoRR*, abs/1909.05855.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, et al. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation.

Semantic Machines, Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676.

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.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. Recipeqa: A challenge dataset for multimodal comprehension of cooking recipes.

Jingyuan Yang, Rongjun Li, and Wei Peng. 2022. ASR Error Correction with Constrained Decoding on Operation Prediction. In *Proc. Interspeech 2022*, pages 3874–3878.

Yichi Zhang, Zhijian Ou, Min Hu, and Junlan Feng. 2020. A probabilistic end-to-end task-oriented dialog model with latent belief states towards semi-supervised learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9207–9219.