Roll Up Your Sleeves: Working with a Collaborative and Engaging Task-Oriented Dialogue System

Lingbo Mo, Shijie Chen[*]**, Ziru Chen**[*]**, Xiang Deng**[†]**, Ashley Lewis**[†]**, Sunit Singh**[†]
Samuel Stevens[†]**, Chang-You Tai**[†]**, Zhen Wang**[†]**, Xiang Yue**[†]**, Tianshu Zhang**[†]**, Yu Su**[‡]**, Huan Sun**[‡]

The Ohio State University

{mo.169, chen.10216, chen.8336, deng.595, lewis.2799, singh.1790, stevens.994, tai.97, wang.9215, yue.149, zhang.11535, su.809, sun.397}@osu.edu

Abstract

We introduce TACOBOT, a user-centered task-oriented digital assistant designed to guide users through complex real-world tasks with multiple steps. Covering a wide range of cooking and how-to tasks, we aim to deliver a collaborative and engaging dialogue experience. Equipped with language understanding, dialogue management, and response generation components supported by a robust search engine, TACOBOT ensures efficient task assistance. To enhance the dialogue experience, we explore a series of data augmentation strategies using LLMs to train advanced neural models continuously. TACOBOT builds upon our successful participation in the inaugural Alexa Prize TaskBot Challenge, where our team secured third place among ten competing teams. We offer TACOBOT as an open-source framework that serves as a practical example for deploying task-oriented dialogue systems.[1]

1 Introduction

Task-Oriented Dialogue (TOD) systems have shown promise in achieving user goals through conversational interactions (Semantic Machines et al., 2020; Su et al., 2022; Mo et al., 2022). However, existing TOD systems focus on users providing information while the system performs tasks. In contrast, our task bot assists users in executing tasks themselves by providing accurate information and guidance.

However, we face several challenges, including the following: (1) Existing TOD systems prioritize functional goals at the expense of user experience. (2) Inadequate in-domain training data, as modern neural models require large amounts of data, and acquiring annotations through crowdsourcing is costly. In this paper, we present TACOBOT, a task-oriented dialogue system designed to assist
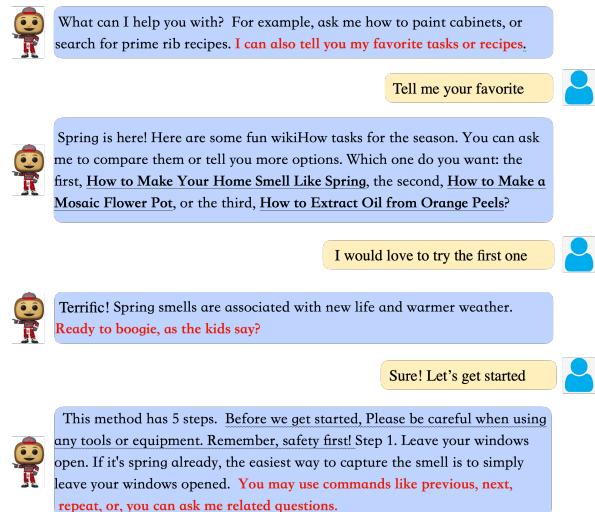

Figure 1: An example dialogue showing first few turns.

users in completing multi-step cooking and how-to tasks. Built upon our previous bot (Chen et al., 2022) deployed in the Alexa Prize TaskBot Challenge (Gottardi et al., 2022), TACOBOT aims to deliver a collaborative and engaging user experience. Figure 1 showcases a partial example dialogue.

Our contributions include: (1) Developing a modularized TOD framework with accurate language understanding, flexible dialogue management, and engaging response generation. (2) Exploring data augmentation strategies, such as leveraging GPT-3 to synthesize large-scale training data. (3) Introducing clarifying questions about nutrition for cooking tasks to personalize search and better cater to user needs. (4) Incorporating chit-chat functionality, allowing users to discuss open topics of interest beyond the task at hand.

2 System Design

2.1 System Overview

TACOBOT follows a canonical pipeline approach for TOD systems. The system consists of three main modules: Natural Language Understanding (NLU), Dialogue Management (DM), and Re-

---

[*] Team co-leads in the challenge with equal contribution.
[†] Other authors in alphabetical order. [‡] Faculty advisors.
[1] Code and datasets are available at OSU-NLP/TacoBot.

sponse Generation (RG). NLU module preprocesses the user's utterance to determine their intent. DM module, designed with a hierarchical finite state machine, controls the dialogue flow, handles exceptions, and guides the conversation towards task completion. RG module generates responses using relevant knowledge and additional modalities to enhance user engagement. Each module is supported by a well-organized knowledge backend and search engine, capable of connecting with various sources to provide optimal user assistance.

2.2 Natural Language Understanding

Our bot employs a robust NLU pipeline which fuses the strengths of pre-trained language models with rule-based approaches. The key component is *Intent Recognition*, where we organize multiple intents into four categories to accommodate a wide array of user initiatives, as detailed in Table 1. Real-world user initiatives often encompass several intents within one single utterance. 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 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.

2.3 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 2.

In the **Task Search phase**, users can search for how-to tasks or recipes directly by issuing a query or ask for task recommendations. TACOBOT retrieves search results from the backend search en-
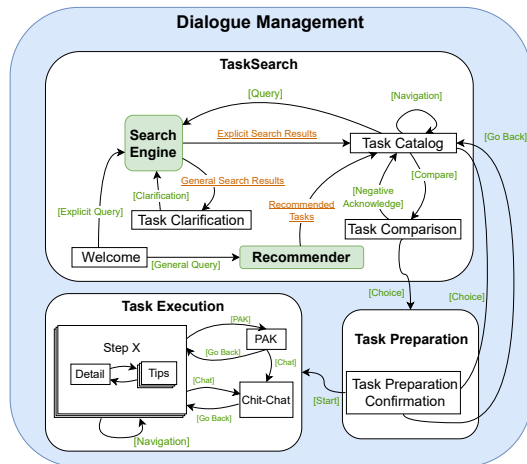

Figure 2: Dialogue Management Diagram. White boxes represent dialogue states and green boxes represent supporting modules. Bidirectional edges represent reflexive transitions. Green texts represent user intent and orange texts denote search engine output.

gine (Section 2.4) and presents candidate tasks for users to compare and select. Once users choose an option, they enter the **Task Preparation phase**. In this phase, users review detailed information about the selected task and decide whether to proceed or search for another task. If users change their mind, they can go back to Task Search and find an alternative task. If they commit to the chosen task, they proceed to the **Task Execution phase**. During this last phase, users follow the step-by-step instructions provided by TACOBOT to complete the task. The utility module, such as the QA module, assists users throughout this phase. Each step of the task has its own state, and for how-to tasks, we break down lengthy steps into shorter instructions, details, and tips for better user comprehension.

DM performs state transitions and selects response generators (Section 2.5) based on user input. The hierarchical design of dialogue states allows for extensible and flexible transitions at different levels. A dialogue state history stack is maintained to facilitate easy navigation to previous states. User intents that do not trigger valid transitions provide contextualized help information to guide users through the dialogue. These design choices ensure stable yet flexible dialogue experiences for users.

2.4 Search Engine

TACOBOT can support diverse tasks backed by large-scale corpus. For the cooking domain, we build a recipe corpus which contains 1.02M recipes based on Recipe1M+ dataset (Marın et al., 2019).

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 1: Categories of detailed intents to support diverse user initiatives.

Meanwhile, we build a wikiHow corpus that includes 93.1K how-to tasks collected from wikiHow website[2]. On top of that, we construct a search engine for both domains based on Elastic search.

2.4.1 Ranking Strategy

To improve the relevance of search results and mitigate the issue of lexical similarity in Elastic search, we employ a query expansion technique that expands user queries by incorporating related words from task names, such as lemmatized verbs, nouns, and decomposed compound nouns. Additionally, we enhance search performance by implementing a neural re-ranking model based on BERT. This model assigns a score to each task by considering the task request and retrieved task titles as input. Training the re-ranker involves employing a weakly-supervised list-wise ranking loss and utilizing synthesized task queries via GPT-3 query simulation. We also propose the collection of weak supervision signals from Google's search engine to avoid the need for human annotation.

2.4.2 Personalized Search

In addition to implementing ranking strategies for accurate search results, our goal is to infuse personalization into the search engine, ensuring a more finely-tuned match with users' needs. To achieve this, we propose a method of asking clarifying questions during recipe searches, collaborating closely with users to understand their preferences regarding nutrition. The logic flow of the process is depicted in Figure 3. Specifically, when a user provides a cooking task of interest, we proactively engage in clarifying discussions with them about the desired level of nutrition in terms of *sugar*, *fat*, *saturates*, and *salt*, using the traffic lights definition established by the Food Standards Agency (FSA).
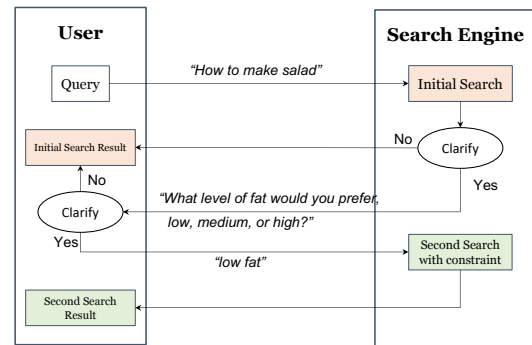
Figure 3: The flow chart for asking clarifying questions.

2.5 Response Generation

Our response generation module blends both infilling-based methods and neural models. We leverage handcrafted conditional rules to organize curated templates and their composition strategy according to the high-level states in our hierarchical finite-state machine. Simultaneously, we build a QA system to respond to diverse user queries.

2.5.1 Question Type Classifier

Our QA system encompasses various question types, including in-context machine reading comprehension (MRC) for context-dependent questions, out-of-context (OOC) QA for open domain questions, frequently-asked questions (FAQ) retrieval for how-to tasks, and rule-based Ingredient and Substitute QA for cooking tasks.

Then, we develop a question type classifier that categorizes user questions into five types (MRC, OOC, FAQ, Ingredient, Substitute) for cooking tasks, and three types (MRC, OOC, FAQ) for how-to tasks. To improve classification accuracy, we concatenate the instruction of the current step (if available) as context with the input question. This combined sequence is then fed into a Roberta-base classifier. Our training set consists of 5,000 questions for each question type, allowing for effective differentiation between different types of questions.

2.5.2 Context-Dependent QA

We begin by annotating an in-context QA dataset comprising 5,183 QA pairs, out of which 752 are unanswerable questions. To ensure reliable responses, we employ Roberta-base to build an extractive QA model in two stages. Initially, we pre-train our model on SQuAD 2.0, followed by fine-tuning on our annotated QA dataset. Recognizing that users may inquire about previously shown steps, we enhance the context by concatenating the current step with the preceding n steps ($n = 2$) during both training and inference processes to prevent information gaps and hallucination.

2.5.3 Context-Independent QA

TACOBOT supports both in-context and context-independent questions. For **out-of-context QA**, we utilize FLAN-T5-XXL (Chung et al., 2022), an instruction-finetuned language model with 11B parameters. Under the zero-shot prompting setup, our bot is equipped to handle open-domain QA and demonstrate commonsense reasoning.

Additionally, **FAQ** module leverages common questions from wikiHow's Community Q&A section, providing answers sourced from real user questions and expert responses. We use a retrieval module based on cosine similarity with question embeddings generated by a sentence-BERT encoder. For **ingredient-related queries**, we employ a high-recall string matching mechanism against the recipe's ingredient list. If users lack a specific ingredient, we suggest alternatives, leveraging a dataset covering 200 commonly used ingredients.

2.6 User Engagement

We develop several strategies to pursue an engaging dialogue experience in the following sections.

2.6.1 Chit-Chat

In real-world conversations, users often desire casual talk alongside the task. To enhance the user experience, TACOBOT offers chit-chat functionality, enabling flexible and diverse conversations. Inspired by Chirpy Cardinal (Chi et al., 2022), we integrate a chit-chat module into our TOD system. A template-based strategy is employed to identify user intent when entering and exiting chit-chat. The chit-chat process consists of three components.

Firstly, **Entity Tracker** monitors entities throughout the conversation, aligning responses with user intentions and focusing on the current topic. Recognized entities allow TACOBOT to access web sources (Wikipedia and Google) and provide intriguing information. Secondly, **Chit-Chat Response Generator** incorporates various response generators: Neural Chat, Categories, Food, Aliens, Wiki, and Transition. Neural Chat uses BlenderBot-3B to generate open-domain responses. Categories and Food generators elicit entity-related responses using templates. Transition facilitates smooth shifts between entities. Wiki enables users to discover engaging information in a conversational style. Aliens presents a five-part monologue series on extraterrestrial existence. Lastly, **Intent Identification Model** determines if the user wants to continue or shift topics. TACOBOT proactively prompts users to return to the task after some chit-chat. Achieving natural transitions between chit-chat and task-oriented dialogue requires ongoing efforts.

2.6.2 People Also Ask

Furthermore, TACOBOT aims to enhance the dialogue experience by delivering captivating content. We leverage Google's "People Also Ask" (PAK) feature, which provides a list of related questions and summarized answers from web pages. This feature reveals popular topics of interest. To collect PAK data, we extract 30k common keywords from task titles in our recipe and wikiHow corpus, resulting in a total of 494k PAK QA pairs.

During task execution, PAK is presented as additional information. To avoid disrupting user focus, we limit the display frequency, currently showing it every 3 steps. Instead of directly displaying the PAK QA pair, we offer an interactive experience by presenting the question first, allowing users to decide if they want to view the corresponding answer. We also provide the option for users to engage in chit-chat if they choose to view PAK.

3 Conclusion

In this paper, we introduce TACOBOT, a modular task-oriented dialogue system that assists users in accomplishing intricate daily tasks. We propose a comprehensive set of modules and approaches to create a collaborative and engaging task bot. To ensure a strong foundation, we employ several data augmentation techniques leveraging LLMs. Furthermore, we open-source the framework and datasets, providing a valuable resource and inspiring future efforts to enhance user-bot collaboration.

Ethics Statement

We present a task bot that is able to converse with users to complete real-world tasks. No personal or identifying information is included throughout conversations. In addition, our bot includes a safety check to ensure safe conversations. We reject inappropriate task requests and prevent showing dangerous tasks, where users and their properties may get hurt. To this end, we perform rule-based matching against a keyword blacklist to filter out inappropriate tasks. Meanwhile, for response generation, we don't directly use LLMs (such as ChatGPT) to generate answers for users' questions, which will have the risk of leaking user data to third-party APIs. Instead, we utilize LLMs to do data augmentation and domain adaptation, and train models locally for the sake of privacy protection.

Author Contributions

In this work, each author makes significant contributions that collectively enhance the final outcome. Lingbo Mo played a crucial role in constructing and organizing the codebase, and building an interactive interface for the demo. Shijie Chen and Ziru Chen co-led the team during the challenge, laying the groundwork for the bot's development. Xiang Deng and Tianshu Zhang were responsible for the NLU pipeline. Xiang Yue mainly developed the QA module. Lingbo Mo and Zhen Wang worked together to build the backend knowledge base and the search engine. Samuel Stevens provided engineering support for constructing an automated test suite. Ashley Lewis focused on enhancing user engagement. Chang-You Tai contributed to chit-chat and PAK features, while Sunit Singh assisted in designing the demo interface. Huan Sun and Yu Su are faculty advisors and offered valuable guidance and feedback.

References

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, Kathleen Kenealy, Swee Kiat Lim, Amelia Hardy, Chetanya Rastogi, Haojun Li, Alexander Iyabor, et al. 2022. Neural generation meets real people: Building a social, informative open-domain dialogue agent. *arXiv preprint arXiv:2207.12021*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Anna Gottardi, Osman Ipek, Giuseppe Castellucci, Shui Hu, Lavina Vaz, Yao Lu, Anju Khatri, Anjali Chadha, Desheng Zhang, Sattvik Sahai, et al. 2022. Alexa, let's work together: Introducing the first alexa prize taskbot challenge on conversational task assistance. *arXiv preprint arXiv:2209.06321*.

Javier Marın, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images.

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.

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.

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.