Multi-Modal Augmentation for Large Language Models with Applications to Task-Oriented Dialogues

Chris Samarinas[*] **Pracha Promthaw, Rohan Lekhwani, Sheshera Mysore,**
Sung Ming Huang, Atharva Nijasure, Hansi Zeng, Hamed Zamani[†]

Center for Intelligent Information Retrieval
Manning College of Information and Computer Sciences
University of Massachusetts Amherst

`{csamarinas, ppromthaw, rlekhwani, smysore, sungminghuan,`
`anijasure, hzeng, zamani}@cs.umass.edu`

Abstract

We introduce MarunaBot V2, an advanced Task-Oriented Dialogue System (TODS) primarily aimed at aiding users in cooking and Do-It-Yourself tasks. We utilized large language models (LLMs) for data generation and inference, and implemented hybrid methods for intent classification, retrieval, and question answering, striking a balance between efficiency and performance. A key feature of our system is its multi-modal capabilities. We have incorporated a multi-modal enrichment technique that uses a fine-tuned CLIP model to supplement recipe instructions with pertinent images, a custom Diffusion model for image enhancement and generation, and a method for multi-modal option matching. A unique aspect of our system is its user-centric development approach, facilitated by a custom tool for tracking user interactions and swiftly integrating feedback. For a demonstration of our system, visit `https://youtu.be/4MNI-puv_eE`.

1 Introduction

Recently, the artificial intelligence landscape has undergone a paradigm shift due to significant advancements in large language models (LLMs). This shift is particularly evident in the field of task-oriented dialog systems (TODS) [19], where systems are evolving to become more competent and user-centric. In this emerging context, we introduce MarunaBot V2, our new system for the second iteration of the Alexa Prize Taskbot Challenge [1]. MarunaBot is designed to assist with daily tasks such as recipes and DIY, leveraging LLMs for data generation and inference to deliver high-quality interactions marked by enhanced robustness and scalability.

The motivation behind this iteration of MarunaBot is to address the unmet needs of users seeking a more engaging and responsive dialogue while navigating complex cooking or DIY tasks. We created a seamless user experience, employing a flexible dialog management logic that can mitigate the effects of unpredictable user behavior to a high extent. Our system has improved rapidly in many iterations thanks to the usage of a custom monitoring tool we developed for systematic error analysis.

We have put substantial emphasis on task search quality. Our four-stage retrieval pipeline delivers high quality and diverse results by fusing many types of signals, including semantic relevance, popularity, rating and rarity of required tools and ingredients.

---

[*] Team Lead [†] Faculty Advisor

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

MarunaBot uniquely incorporates multi-modality, aiming to augment user interaction by integrating diverse forms of information such as text, images, and videos from multiple sources. We have also developed models for text-based image retrieval, image generation and enhancement in order to further enrich and improve our corpus. A unique feature of our system is visual option matching, which allows users to select recipes based on their visual features.

2 System Overview

The architecture of our system is summarized in Figure 1. We have constructed our taskbot utilizing the Cobot Toolkit framework, which has significantly reduced our engineering time, allowing us to concentrate primarily on research. The interaction between our taskbot and the user is facilitated through a multi-modal or headless Alexa device. We process a variety of user signals, including speech, text, and touch events. Once these input signals are transformed into text or action, they are forwarded to the core component, the dialogue manager (DM). The DM interacts with two hybrid models to determine the appropriate response generator for the given user input, namely the intent classifier and the option matching. We have developed 21 distinct response generators that the DM utilizes, including those for presenting search results, displaying step information, answering questions, and engaging in chitchat (see Figure 2 for an example and the appendix for the whole list). The question answering and chitchat response generators interact with two external models. For option matching and intent classification, we employ an ensemble approach that leverages two types of large language models, FlanT5 [4] and ChatGPT [14].

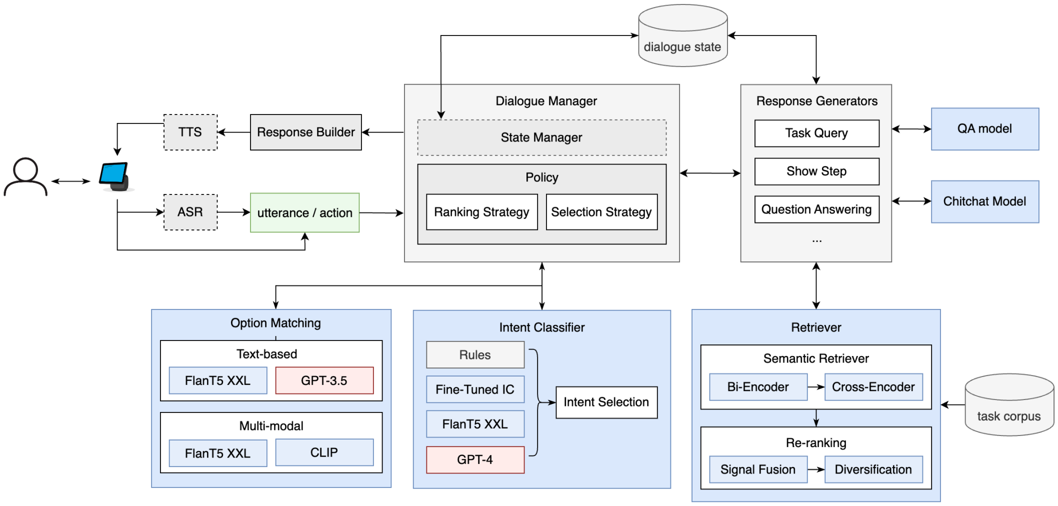
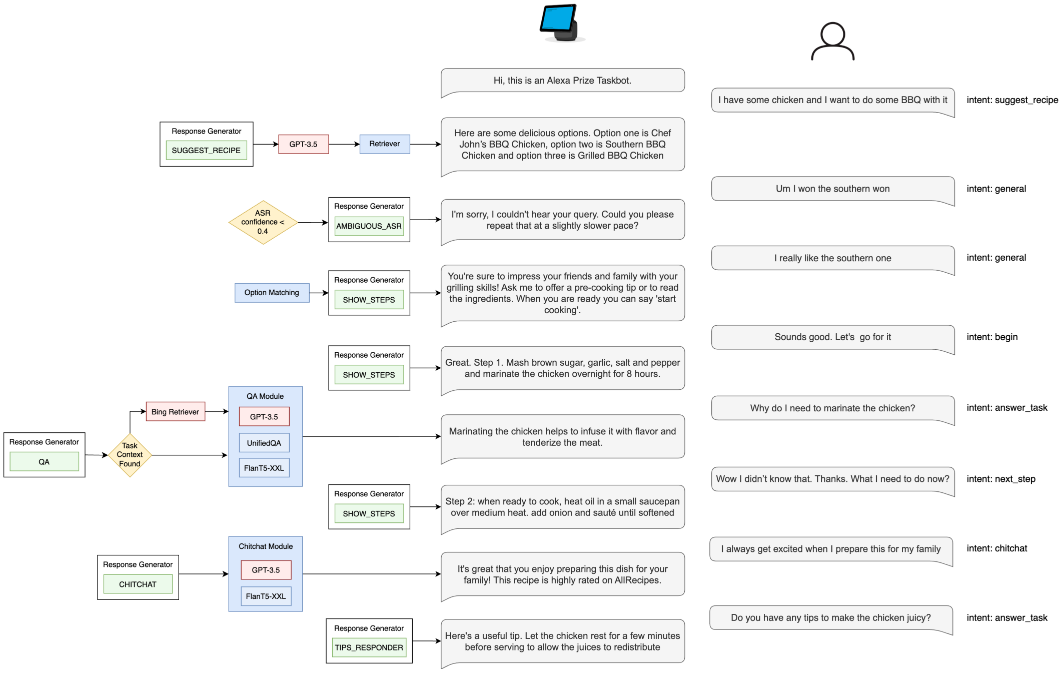


Figure 1: The architecture of MarunaBot V2. The components enclosed by dashed borders are those provided by the Cobot Toolkit and used without modifications.

In the following sections, we present our approach to corpus construction, our models and research findings for various tasks including intent classification, task retrieval, question answering, and multi-modality. We discuss our user-driven development approach and finally, we present our preliminary results from our end-to-end retrieval-augmented taskbot using a custom large language model.

3 Intent Classification

Accurate and robust intent classification is a critical aspect of task-oriented dialogue systems, as it dictates the flow of interaction between the user and the system. Misclassification often leads to irrelevant or unsatisfactory responses, negatively impacting the user experience. To address this, our system is designed to accommodate a broad spectrum of user requests, including those that are out-of-domain or potentially unsupported. We have defined a comprehensive taxonomy of 36 intents to enable a more sophisticated user-system interaction (see Table 9 in appendix). These intents range from navigation and content filtering to domain classification, ambiguous requests, and out-of-domain



Hi, this is an Alexa Prize Taskbot.

I have some chicken and I want to do some BBQ with it

Here are some delicious options. Option one is Chef John's BBQ Chicken, option two is Southern BBQ Chicken and option three is Grilled BBQ Chicken

Um I won the southern won

I'm sorry, I couldn't hear your query. Could you please repeat that at a slightly slower pace?

I really like the southern one

You're sure to impress your friends and family with your grilling skills! Ask me to offer a pre-cooking tip or to read the ingredients. When you are ready you can say 'start cooking'.

Sounds good. Let's go for it

Great. Step 1. Mash brown sugar, garlic, salt and pepper and marinate the chicken overnight for 8 hours.

Why do I need to marinate the chicken?

Marinating the chicken helps to infuse it with flavor and tenderize the meat.

Wow I didn't know that. Thanks. What I need to do now?

Step 2: when ready to cook, heat oil in a small saucepan over medium heat. add onion and sauté until softened

I always get excited when I prepare this for my family

It's great that you enjoy preparing this dish for your family! This recipe is highly rated on AllRecipes.

Do you have any tips to make the chicken juicy?

Here's a useful tip. Let the chicken rest for a few minutes before serving to allow the juices to redistribute

Figure 2: An example conversation (not from real user), the identified intents and the called response generators in each case.

inquiries. Certain intents require slot-filling (e.g. search intents, ingredient substitution, and timer setting), which is performed jointly using a single model.

In-Context Learning The lack of abundant training data presented a significant challenge. To mitigate this, we adopted a few-shot learning approach with in-context learning [5], utilizing FlanT5-XXL [4]. We created a prompt that included input and output few-shot examples for all intents. However, this initial approach was not sufficiently robust and exhibited high latency, reaching up to 2 seconds due to the lengthy prompt (629 tokens).

Retrieval Augmentation To decrease inference time, we experimented with reducing the prompt length without compromising its effectiveness. We employed a pre-trained transformer-based bi-encoder[3] to retrieve the most relevant few-shot examples corresponding to the given input utterance. This strategy successfully reduced the average inference time from 1.6 seconds to 0.5 seconds using the three closest few-shot examples, while simultaneously improving the macro F1 score by 2%. The effectiveness of the retrieval augmentation strategy can be seen in Figure 3.

Synthetic Data Generation For this task we relied on synthetic data generation using ChatGPT. We investigated the usage of real user utterances from past conversations as training data after manual and automatic annotation with GPT-4, but we observed lack of diversity and infrequent representation of many classes, at least in the first months of the Taskbot Challenge. For this reason, we relied only on automatically generated user utterances. For each intent class, we constructed a prompt that encapsulated the role/setting (user interacting with a taskbot assisting with recipes and tasks), a succinct description of the intent, and 3-5 seed examples. We utilized ChatGPT (versions 3.5 and 4) to generate 2000 (1000 from each model version) synthetic utterance examples for each intent class, with temperature values ranging from 0.7 to 1.0, contingent on the intent type. Near-duplicate utterances were eliminated using nearest-neighbor retrieval with a pre-trained text bi-encoder[3] and empirically defined dot product threshold.

Experimental Results We curated a test set manually, comprising 785 examples that covered all the defined intents. Using this set, we computed precision, recall, and F1 for each class, as well as macro F1 as our aggregate metric. Using the synthetic data, we fine-tuned FlanT5, DeBERTa, and mp-net

---

[3] https://hf.co/sentence-transformers/all-mpnet-base-v2

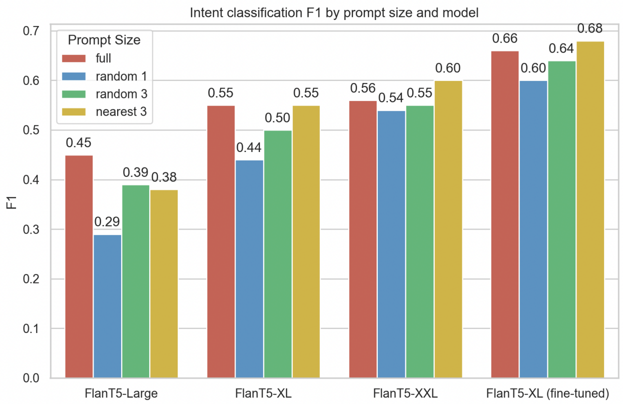


Figure 3: Few-shot intent classification results with top-3 retrieval augmentation.

Model	**Accuracy**	**Macro F1**
FlanT5-XXL (prompting)	0.68	0.59
FlanT5-XL (fine-tuned)	**0.76**	**0.66**
DeBERTa (fine-tuned)	0.55	0.48
mpnet-base-v2 (contrastive fine-tuning)	0.66	0.57
GPT-4 (prompting)	0.74	0.65

Table 1: Comparison of various models for intent classification.

bi-encoder [3] with contrastive distance-dependent loss [7]. Our FlanT5-XL model outperformed the other fine-tuned models. We also compared our model with GPT-4 using the few-shot prompt and found performance to be very similar to the fine-tuned FlanT5-XL. The results can be seen in Table 1. For more detailed results per intent you can refer to the appendix. Interestingly, GPT-4 exhibited greater robustness in practice, suggesting that our curated test set may not fully evaluate the robustness of our best fine-tuned model. This finding prompts questions about the design of datasets for evaluating complex tasks that are prone to high variability and noise in real-world scenarios.

Hybrid Approach Our final approach to intent classification is a hybrid one, incorporating zero-shot FlanT5 & GPT-4, fine-tuned FlanT5, and rule-based methods. Rules are given the highest priority, followed by the fine-tuned FlanT5-Large, and then a concurrent call of FlanT5-XXL and GPT-4. The latter two are only invoked for infrequent intents due to their increased latency. For instance, for navigation-related intents (next step, acknowledge), only the efficient FlanT5-Large model is utilized. When calling FlanT5-XXL and GPT-4, we prioritize the output of GPT-4 if a timeout constraint of 4 seconds is met. This hybrid approach allows us to harness the strengths of different models, ensuring a more robust and efficient intent classification system.

4 Task Retrieval

The retrieval system of MarunaBot V2 handles the pivotal task of sourcing the appropriate recipe or DIY instructions in response to a user's request. This complex process was addressed through careful data gathering, optimization of the retrieval models, and utilization of LLMs for for complex queries.

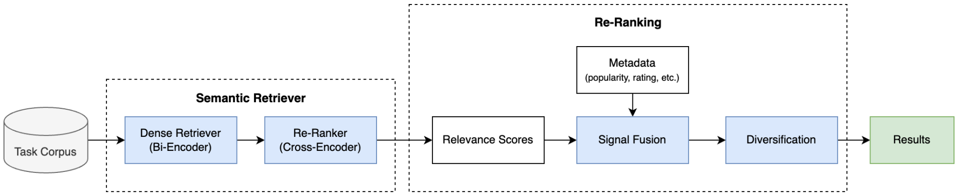


Figure 4: Overview of the task retrieval pipeline in MarunaBot V2.

Corpus Construction We compiled a comprehensive corpus from various sources, resulting in a unified corpus encompassing 81,593 tasks and recipes (16k recipes and 65k tasks). Prior to inclusion, potential content underwent a rigorous screening process with GPT-3.5 to filter out any inappropriate information, such as medical, legal, financial advice, or dangerous tasks. LLMs, particularly GPT-4 and GPT-3.5, were employed to curate and generate considerable metadata (including tips, tools, positive statements, fun facts, summaries, related tasks) to enrich the user's experience.

Retrieval Pipeline Our retrieval pipeline (Figure 4) fuses into one result quality score two types of signals; relevance and attribute-related. The relevance signal is supplied by the semantic retriever after a two stage process using a pre-trained bi-encoder followed by a fine-tuned cross-encoder (more details in section 4.1). The attribute-related signals include popularity, rating, difficulty, rarity of ingredients or tools, and popularity estimates from the web. The final stage of the pipeline is diversification of the results using Maximal Marginal Relevance (MMR) [3]. This ensures that there is not a lot of repetition in the first presented results to the user.

Grounded LLM-based Results For more complex queries that not explicitly mention specific recipe names, such as queries related to occasions or personal preferences, GPT-3.5 was instrumental in suggesting appropriate recipes and recommending related results in the end of each task. The LLM-based suggestions were further grounded to our corpus using our custom retrieval model described in the following section.

4.1 Retrieval Models

We employ a two-stage retrieval pipeline using Transformer language model biencoders for initial retrieval and cross-encoders for re-ranking. Synthetic queries for training are generated from a collection of documents using GPT-3.5 and GPT-3.

Synthetic data generation We sample 60k wikihow documents and recipes from our 300k document collection, generating three synthetic queries per document using a mixture of GPT-3.5 and GPT-3, resulting in 180k positive pairs. The queries exhibit different characteristics, with GPT-3 queries being more extractive and ChatGPT queries requiring social and common sense knowledge [2, 13].

Retrieval models Bi-encoder and cross-encoder models are trained on synthetic data using a pre-trained transformer language model, MPNET [18]. Bi-encoder models embed the query and item independently into vectors and rank items based on the minimum L2 distance. Cross-encoder models input both the query and item and output a score for ranking. We train our models with margin ranking loss and cross-entropy loss, respectively. In later section, we will called the fine-tuned Bi-encoder and Cross-encoder as `BiEnc`-Syn, and `CrEnc`-Syn respectively.

Experimental Setup We evaluate our models on a unified collection of 300k wikihow and recipe documents. `BiEnc`-Syn retrieves documents from the whole collection, while `CrEnc`-Syn re-ranks the top 30 documents from a baseline bi-encoder MPNet-250QA. Evaluation is done using a held-out set of synthetic queries and their corresponding documents, following the 1-shot automatic relevance estimation method of MacAvaney and Soldaini [12]. We compare our approach to several bi-encoder and cross-encoder baselines, including Contriver, MPNet-1B, BERT-MSM, MPNet-250QA, and MiniLM-MSM [9, 17].

Results Our trained bi-encoder underperforms the best prior biencoders, likely due to a smaller training set and lack of optimized negative samples (Table 2). However, our trained `CrEnc`-Syn outperforms several baseline models. We also ablate several design choices (Table 3), finding that using description text improves results, re-ranking top 30 documents outperforms re-ranking top 200, and MPNet-250QA outperforms the MPNet-1B model as a first stage ranker.

5 Question Answering

Our approach for question answering is a text-to-text model fine-tuned for QA. Given an input question and passage (e.g. recipe or DIY instructions), our approach generates an answer from the passage or identifies that the question cannot be answered from the passage. Our model is fine-tuned on questions and answers synthetically generated from GPT-3.5.

Synthetic Data Generation Since we are interested in training a model for QA while also detecting when the question is unanswerable from a passage, we generate synthetic answerable question-

Description	Model	P@5	P@10	P@20
Off-the-shelf Biencoder	MPNet-1B	0.7165	0.6926	0.6769
	MPNet-250QA	*0.7250*	*0.6988*	*0.6790*
	BERT-MSM	0.7132	0.6897	0.6731
	Contriver	0.6978	0.6806	0.6675
Fine-tuned Biencoder	`BiEnc-Syn`	0.7114	0.6915	0.6764
Off-the-shelf Crossencoder	MiniLM-MSM	0.7354	0.7079	0.687
Fine-tuned Crossencoder	`CrEnc-Syn`	**0.7422**	**0.715**	**0.6911**

Table 2: Performance of pre-trained biencoders and cross-encoders compared against the cross-encoder trained on synthetic queries generated for DIY and Recipe documents. Both cross-encoders use MPNet-250QA biencoders for first-stage retrieval and re-rank the top 30 documents.

Description	Model	P@5	P@10	P@20
Best system	`CrEnc-Syn`	**0.7422**	**0.715**	**0.6911**
Title text	MiniLM-MSM	0.7173	0.6954	0.6750
Description text	MiniLM-MSM	0.7314	0.7049	0.6836
Re-rank Top-30	MiniLM-MSM	0.7354	0.7079	0.6870
Re-rank Top-200	MiniLM-MSM	0.7286	0.7062	0.6887
Biencoder: MPNet-1B	MiniLM-MSM	0.7314	0.7049	0.6836
Biencoder: MPNet-250QA	MiniLM-MSM	0.7354	0.7079	0.6870

Table 3: Ablation experiments indicating the impact of design choices for cross-encoders. Our best system uses a MPNet-250QA, re-ranks 30 documents, and uses description in addition to the title.

answer pairs and unanswerable questions. To generate synthetic data for fine-tuning, we select 15,000 passages spanning DIY and Recipes. Then, ChatGPT is instructed to generate ten answerable questions and their answers from the passage and ten unanswerable questions. This results in 300k question-passage pairs for fine-tuning.

Models Our models for QA follow the T5 model architecture of varying sizes (770M and 3B parameters) [16]. Our models are initialized from UnifiedQA models, T5 models fine-tuned on 20 different QA datasets and found to be a good initialization for several different QA benchmarks [10, 11]. We further fine-tune UnifiedQA for DIY and recipe question answering.

Experimental Setup Our evaluation data is a random sample of 300 passages paired with their question-answer pairs. Answerable question-answer pairs and unanswerable questions are synthetically generated and treated as correct – in manual examination the question answer pairs were often found to be correct. Given that the answers in our test set are fluent natural language sentences we used BertScore [20], a metric for evaluating text generation models by comparing generated texts to a reference text, to compute evaluation metrics.

Baselines We compare our proposed models to the following zero-shot QA baselines, FlanT5 and UnifiedQA. FlanT5 models are T5 models finetuned for instruction following and demonstrating strong zero-shot performance on numerous benchmarks [4]. To refrain from generating answers for unanswerable questions we instruct the model only to generate an answer where it is possible from the passage. We experiment with model variants with 770M parameters and 3B parameters. UnifiedQA models are performant QA models pre-trained for QA on 20 different QA datasets [10, 11]. Notably, these models cannot determine when questions are unanswerable from the provided passage.

Results Our proposed fine-tuned models significantly outperform baseline approaches based on FlanT5 and UnifiedQA on both answerable and unanswerable questions. Across model sizes, a UnifiedQA model outperforms a FlanT5 model. While in zero-shot performance, the FlanT5 model can detect unanswerable questions significantly better than a UnifiedQA model the UnifiedQA model can generate more accurate answers for answerable questions. After fine-tuning, UnifiedQA models

Training	Model	Parameters	All	Unanswerable	Answerable
Zero-shot	FlanT5-Large	770M	62.34	77.54	47.15
	FlanT5-XL	3B	61.64	72.61	50.67
	UnifiedQA-Large	770M	30.94	07.52	54.36
	UnifiedQA-XL	3B	31.90	07.42	56.39
Fine-tuned	FlanT5-Large	770M	74.90	97.74	52.06
	UnifiedQA-Large	770M	*80.47*	*92.43*	*68.52*
	UnifiedQA-XL	3B	**84.78**	**96.57**	**72.99**

Table 4: Model performance for question answering in recipe and DIY passages. Metrics represent macro-averages over questions in the test set. While "Answerable" reports the correctness of generated answers, "Unanswerable" merely reports the ability of a model to detect an unanswerable question. Performance is measured in F1 based on BERTSCORE based on a reference answer.

outperformed FlanT5 models. While Large models sometimes match or outperform XL models, XL models generally outperform Large models in the case of answerable questions; this trend holds across FlanT5 and UnifiedQA models in zero-shot and fine-tuned setups.

6 Multimodal Integration

Task-oriented Dialogue Systems can significantly benefit from the integration of multiple communication modes, enriching the interaction and enhancing user experience. This section elaborates on how we augmented our system with images using both retrieval and generative methods.

6.1 Retrieval-based Image Augmentation

We utilized two pre-trained CLIP models [15], CLIP-ViT-L14 and CLIP-ViT-B32, to conduct text-based image retrieval experiments. Initially, we performed zero-shot image retrieval on the WikiHow dataset's embedding using both models. This was followed by data retrieval from the same dataset, based on the embedding created from text and image features. We then fine-tuned both models using our custom training data, which consisted of recipe-image-text pairs. The models underwent training following the hyperparameter selection from Dong et al. [6]. We repeated the experiments using these fine-tuned models and evaluated their performance using Mean Reciprocal Rank. We selected 25 sample queries, each representing a general recipe step. For each query in the WikiHow Dataset, we ranked the top 10 images.

Dataset Collection We assembled the WikiHow recipe image-text dataset and employed a classifier to determine whether an image represented a recipe. We also collected frames from over 40K recipe step videos, using a CLIP_L14 threshold between frame and step text to enhance dataset quality. A threshold score of 21 was used to ensure the quality of image-text matching. The text was the recipe text associated with the step video. In total, we created 231k recipe-text image pairs.

The fine-tuned CLIP-ViT-B32 model demonstrated superior performance in retrieving relevant images. The results are presented in Table 5. We observed improved retrieval with the fine-tuned model, where the images were more closely aligned with the steps (Figure 5 and more examples in the appendix).

Model	Input	MRR
clip-ViT-B-32 (zero-shot)	image	0.1196
clip-ViT-B-32 (zero-shot)	text + image	0.1973
clip-ViT-B-32 (fine-tuned)	image	0.3151
clip-ViT-B-32 (fine-tuned)	text + image	0.3764

Table 5: Experiments with CLIP for text-based recipe step image retrieval.

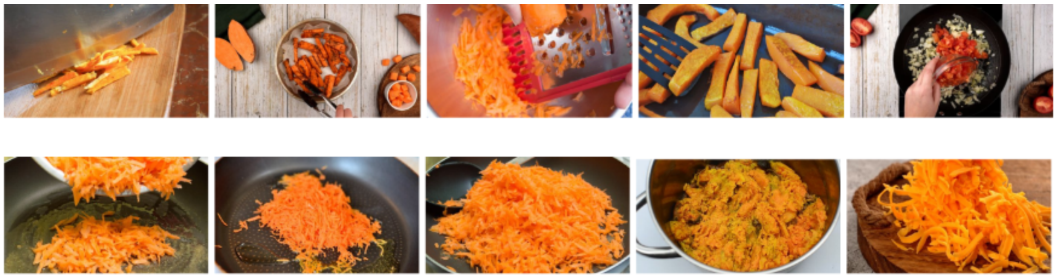


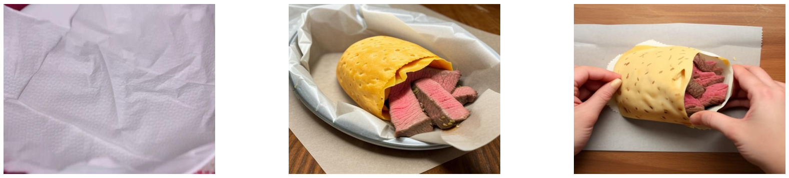
Figure 5: The top 5 retrieved images for the query 'add shredded sweet potato and cook 1 to 2 minutes or until it begins to stick to the pan.' using zero-shot CLIP (top) and fine-tuned (bottom).

6.2 Generative Image Augmentation

Initially, we compiled a dataset of text-image pairs from our recipe steps corpus. We then used ChatGPT (GPT3.5) to generate context-independent text, which served as the foundation for comprehensive image generation. Concurrently, we calculated the CLIP scores for these pairs and selected the 100 pairs with the lowest scores, excluding context-dependent text. Then we fine-tuned a Stable Diffusion model using LoRA [8] and the compiled corpus for recipe step image generation. We generated images under various model configurations using these refined pairs. To compare the performance of the generated images with the retrieved ones, we computed the CLIP scores. We further nuanced this evaluation by experimenting with different LoRA weights.

Models Initially, we retrieved images from the corpus using the CLIP model. We then employed the Stable Diffusion model, using the Euler ancestral sampling method, and a model checkpoint of RealisticVision V3.0.[4] We also explored the implementation of the Stable Diffusion model with our fine-tuned LoRA for recipe step image generation. Here, we experimented with varying weights to study their impact on the model's performance. The weights under consideration ranged from 0.4 to 1.0. This approach provided insights into the influence of LoRA's weight on image quality.

Results As per the results in Table 6, we found that the generated image significantly improved the CLIP score compared to the retrieved image when the CLIP score was below 21. However, there was no substantial difference in CLIP scores between the image generated with stable diffusion alone and the one produced with LoRA. As depicted in Figure 6, the model incorporating LoRA tended to generate images with a more consistent format, such as identical angles, and images leaning towards action steps rather than post-cooking photos. The presence of noise within the training data, such as text, low-resolution, and moving images, could potentially account for the resultant images lacking the resolution and aesthetic quality of the original model. More results can be found in Figure 14 in the appendix.



(a) Retrieved with CLIP (b) Generated (c) Generated with LoRA

Figure 6: Images for 'Scoop 1/4 cup of the beef and cheddar filling onto the wrapper.'

6.3 Enhancing Image Aesthetics

Our approach for image enhancement employs an image-to-image generation model. By inputting an image and a prompt, we aim to generate an image that is visually more appealing while preserving its original structure. The model incorporates the Stable Diffusion Model (Realistic Vision V4) in conjunction with depth-based Controlnet and a custom LoRA.[5]

---

[4] https://huggingface.co/SG161222/Realistic_Vision_V3.0_VAE

[5] https://civitai.com/models/45322/food-photography

Model	Mean	Std.
Retrieved Image (baseline)	16.508	3.286
Generated Image + LoRA (weight = 0.4)	24.765	4.562
Generated Image	25.266	4.751

Table 6: CLIP score for image generation comparing to image retrieval approach

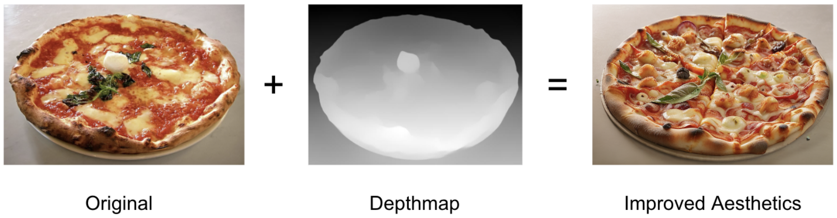


Figure 7: Improving aesthetics of a recipe image using Stable Diffusion and ControlNet

Enhancement Process We formulate the prompt for image generation by combining the recipe title with its ingredients. This approach is adopted due to the observed limitations when using only the title for image generation. For instance, when provided with the title "apple pie", the model generated an image of a pie with a whole apple in the middle, which was not as per our expectations. However, when the ingredients were incorporated into the prompt, the quality of the generated image improved significantly.

Evaluation We sample 1k item from our recipe corpus, and calculate their aesthetic score with CLIP. We pick the 100 item of the lowest aesthetic score and generate the images for each these items. After we get enhanced image, we calculate their CLIP score and aesthetic score. The scores are in below Table. We can see a significant improvement over the aesthetics while preserving the relevance of the generated image.

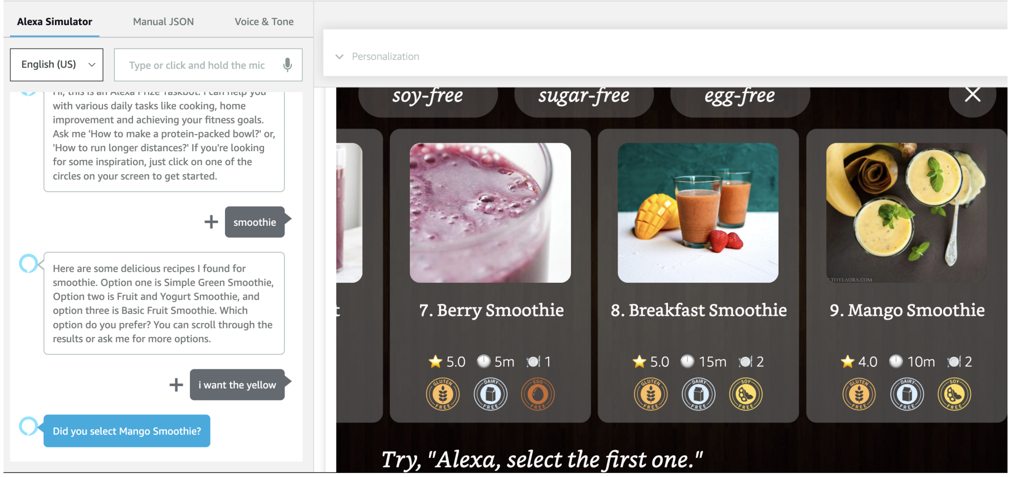
Type	Enhanced	Score
CLIP Score (0 to 1)	No	0.3021
	Yes	0.3040
Aesthetic Score (out of 10)	No	4.933
	Yes	5.983

Table 7: Relevance evaluation of generated images using CLIP

The investigation centers on the utilization of recipe titles as the basis for scoring the relationship between text and image, acknowledging that such titles may not comprehensively convey the visual appearance of dishes. Consequently, the CLIP score did not exhibit significant improvements as expected. Conversely, the aesthetic evaluations yielded noteworthy enhancements, indicating a substantial improvement in the overall visual quality of the represented images.

6.4 Visual Option Matching

Visual option matching is a unique experimental feature in our system that allows the user to select a presented recipe based on its visual appearance. Given an text query and the image of a given recipe, we first extract the recipe name from the user's utterance using FlanT5-XXL through prompting. Then we try to identify a potential match between the given recipe name and the displayed images with dot-product similarity. If a match above a threshold is found, we ask a clarifying question to the user if they indeed selected that option (see Figure 8).



Hi, this is an Alexa Prize Taskbot. I can help you with various daily tasks like cooking, home improvement and achieving your fitness goals. Ask me 'How to make a protein-packed bowl?' or, 'How to run longer distances?' If you're looking for some inspiration, just click on one of the circles on your screen to get started.

smoothie

Here are some delicious recipes I found for smoothie. Option one is Simple Green Smoothie, Option two is Fruit and Yogurt Smoothie, and option three is Basic Fruit Smoothie. Which option do you prefer? You can scroll through the results or ask me for more options.

i want the yellow

Did you select Mango Smoothie?

Figure 8: Example of visual match.

7 User-driven Development

The development of MarunaBot v2 followed a user-centric design approach. The system was engineered with an anticipation of unpredictable user behavior, and strong emphasis on multimodal device utility. In scenarios where user requests prove ambiguous, unclear, or outside the system's scope of support, MarunaBot is capable of generating appropriate responses that provide guidance and clarifying the extents of its functional capabilities.

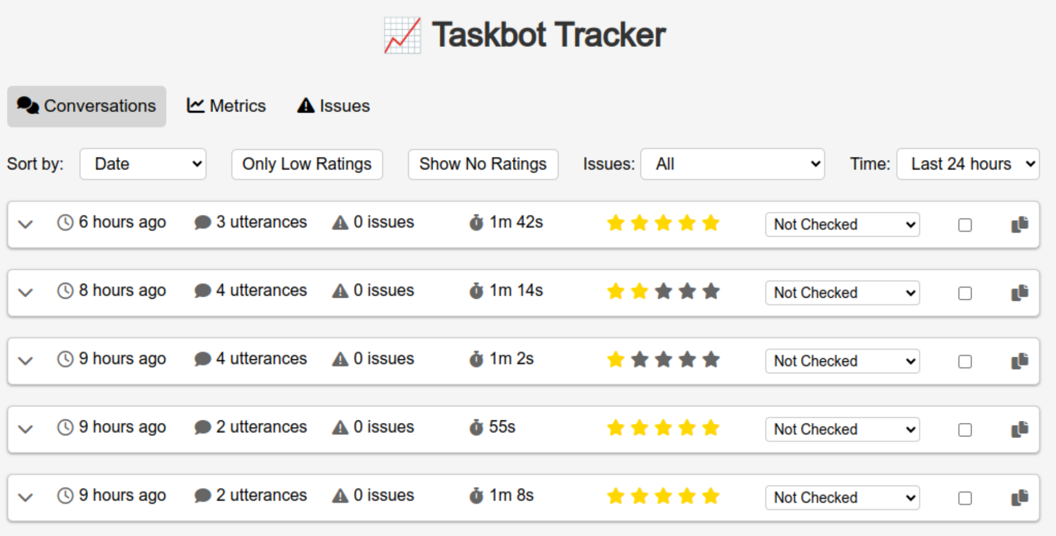
To enable this process, we developed a customized monitoring tool that provides a daily review of user conversations and received feedback (see figures 9, 10 , and 11). This tool offered invaluable insights by identifying patterns and common issues that were hindering the system's usability. The tool further enabled the visualization of an array of performance metrics over time influenced by this user-driven development process, including the total number of conversations, identified issues, overall user ratings, interaction durations, and the number of dialogue turns. The monitored conversations also served as a form of continuous user feedback, aiding in the further fine-tuning of our intent classifier. On a daily basis, the incorporation of real user utterances into our training data contributed to honing our taskbot's ability to correctly identify user intent and respond appropriately.

An experimental feature of our monitoring tool was implemented in the form of a rating estimate mechanism driven by a language model. A DeBERTa-v3-large model was trained on 883 rated user conversations with an input combined of user utterances, system responses, the intent class and sentiment of the user's utterance. Ratings greater than or equal to three were classified as a 1 and those less than three as a 0. With this setup, our model achieved 0.7063 F1 score, with precision 0.7163 and recall 0.6966. Although the ratings assigned by the users were not always correlated to their conversation quality with the bot, this approach yielded promising early results considering not just what the user said but also how they said it using sentiment information.

Moreover, we conducted initial experiments with GPT-4 to detect specific issues in user conversations. However, in compliance with our commitment to user privacy, these models were not deployed into production environment, marking this as an area of potential exploration for future research.

8 Conclusion & Future Work

In this paper, we introduced MarunaBot V2, an advanced Task-Oriented Dialogue System designed to assist users in cooking and Do-It-Yourself tasks. Our system leverages large language models for data generation and inference, and implements hybrid methods for intent classification, retrieval, and question answering, thereby achieving an optimal balance between efficiency and performance. A distinguishing feature of our system is its multi-modal capabilities, which include a multi-modal enrichment technique using a fine-tuned CLIP model, a custom Diffusion model for image enhancement and generation, and a method for multi-modal option matching. Our user-centric development

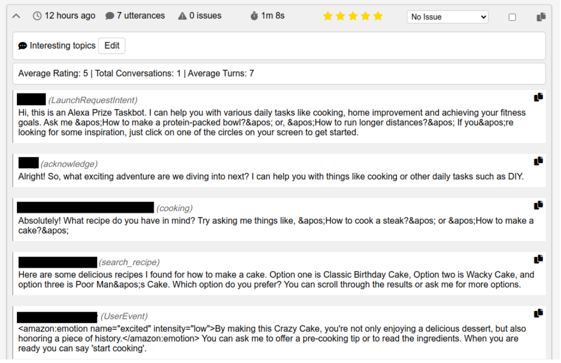


Taskbot Tracker

Sort by: Date | Only Low Ratings | Show No Ratings | Issues: All | Time: Last 24 hours

6 hours ago	3 utterances	0 issues	1m 42s	★★★★★	Not Checked	☐	
8 hours ago	4 utterances	0 issues	1m 14s	★★☆☆☆	Not Checked	☐	
9 hours ago	4 utterances	0 issues	1m 2s	★☆☆☆☆	Not Checked	☐	
9 hours ago	2 utterances	0 issues	55s	★★★★★	Not Checked	☐	
9 hours ago	2 utterances	0 issues	1m 8s	★★★★★	Not Checked	☐	

Figure 9: Taskbot Tracker: Tool for tacking user interactions and swiftly integrating feedback.



12 hours ago | 7 utterances | 0 issues | 1m 8s | ★★★★★ | No Issue | ☐

Average Rating: 5 | Total Conversations: 1 | Average Turns: 7

[redacted] (LaunchRequestIntent)
Hi, this is an Alexa Prize Taskbot. I can help you with various daily tasks like cooking, home improvement and achieving your fitness goals. Ask me &apos;How to make a protein-packed bowl?&apos; or, &apos;How to run longer distances?&apos; If you&apos;re looking for some inspiration, just click on one of the circles on your screen to get started.

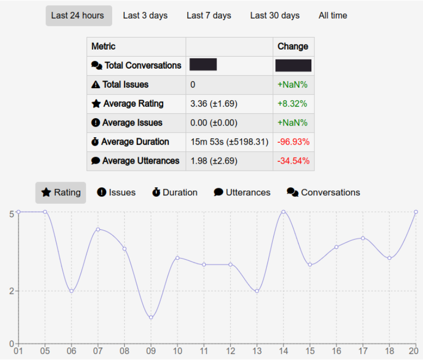
[redacted] (acknowledge)
Alright! So, what exciting adventure are we diving into next? I can help you with things like cooking or other daily tasks such as DIY.

[redacted] (cooking)
Absolutely! What recipe do you have in mind? Try asking me things like, &apos;How to cook a steak?&apos; or &apos;How to make a cake?&apos;

[redacted] (search_recipe)
Here are some delicious recipes I found for how to make a cake. Option one is Classic Birthday Cake, Option two is Wacky Cake, and option three is Poor Man&apos;s Cake. Which option do you prefer? You can scroll through the results or ask me for more options.

[redacted] (UserEvent)
<amazon:emotion name="excited" intensity="low">By making this Crazy Cake, you're not only enjoying a delicious dessert, but also honoring a piece of history.</amazon:emotion> You can ask me to offer a pre-cooking tip or to read the ingredients. When you are ready you can say 'start cooking'.

Figure 10: Conversation within the Taskbot Tracker also shows average lifetime rating given by a user, total conversations with the bot and the average turns.



Metric		Change
Total Conversations	[redacted]	[redacted]
Total Issues	0	+NaN%
Average Rating	3.36 (±1.69)	+8.32%
Average Issues	0.00 (±0.00)	+NaN%
Average Duration	15m 53s (±5198.31)	-96.93%
Average Utterances	1.98 (±2.69)	-34.54%

Figure 11: Crucial bot metrics can be monitored easily using the Taskbot Tracker.

approach, facilitated by a custom tool for tracking user interactions and swiftly integrating feedback, sets our system apart.

There are many areas where our system can improve. We identified the need to refine our approach for generative image augmentation, specifically by filtering out frames or images containing text or moving elements that could introduce noise into the generated image. We also recognized the challenge of accurate intent classification. Future work will focus on enhancing the intent classification model's ability to predict correct context-dependent intents by classifying user utterances based on sentiment and the user's position within the conversation.

Acknowledgments. We would like to thank Rahul Seetharaman for some early contributions to our taskbot. This work was supported in part by the Center for Intelligent Information Retrieval, in part by an Amazon Alexa Prize grant, in part by NSF grant number 2143434, and in part by the Office of Naval Research contract number N000142212688. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

References

[1] E. Agichtein, M. Johnston, A. Gottardi, C. Flagg, L. Vaz, H. Shi, D. Zhang, L. Ball, S. Liu, L. Dai, D. Pressel, P. Goyal, L. Hu, O. Ipek, S. Sahai, Y. Lu, Y. Liu, D. Hakkani-Tür, S. Hu, H. Rocker, J. Jeun, A. Iyengar, A. Mandal, S. Kuzi, N. Vedula, O. Rokhlenko, G. Castellucci, J. I. Choi, K. Bland, , Y. Maarek, and R. Ghanadan. Alexa, let's work together: Introducing the second alexa prize taskbot challenge. In *Alexa Prize TaskBot Challenge 2 Proceedings*, 2023. URL `https://www.amazon.science/alexa-prize/proceedings/alexa-lets-work-together-introducing-the-second-alexa-prize-taskbot-challenge`.

[2] L. Bonifacio, H. Abonizio, M. Fadaee, and R. Nogueira. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2387–2392, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531863. URL `https://doi.org/10.1145/3477495.3531863`.

[3] J. G. Carbonell and J. Goldstein-Stewart. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[4] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei. Scaling instruction-finetuned language models, 2022.

[5] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui. A survey on in-context learning, 2023.

[6] X. Dong, J. Bao, T. Zhang, D. Chen, S. Gu, W. Zhang, L. Yuan, D. Chen, F. Wen, and N. Yu. Clip itself is a strong fine-tuner: Achieving 85.7% and 88.0% top-1 accuracy with vit-b and vit-l on imagenet. *arXiv preprint arXiv:2212.06138*, 2022.

[7] A. Hattimare, A. Dharawat, Y. Khan, Y.-C. Lien, C. Samarinas, G. Z. Wei, Y. Yang, and H. Zamani. Maruna bot: An extensible retrieval-focused framework for task-oriented dialogues. In *1st Proceedings of Alexa Prize TaskBot (Alexa Prize 2021)*, 2022.

[8] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models, 2021.

[9] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL `https://openreview.net/forum?id=jKN1pXi7b0`.

[10] D. Khashabi, S. Min, T. Khot, A. Sabharwal, O. Tafjord, P. Clark, and H. Hajishirzi. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.171. URL `https://aclanthology.org/2020.findings-emnlp.171`.

[11] D. Khashabi, Y. Kordi, and H. Hajishirzi. Unifiedqa-v2: Stronger generalization via broader cross-format training. *arXiv preprint arXiv:2202.12359*, 2022.

[12] S. MacAvaney and L. Soldaini. One-shot labeling for automatic relevance estimation. *arXiv preprint arXiv:2302.11266*, 2023.

[13] S. Mysore, A. McCallum, and H. Zamani. Large language model augmented narrative driven recommendations. *RecSys*, 2023.

[14] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022.

[15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435.

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

[18] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c3a690be93aa602ee2dc0ccab5b7b67e-Paper.pdf.

[19] H. Zamani, J. R. Trippas, J. Dalton, and F. Radlinski. Conversational information seeking. *ArXiv*, abs/2201.08808, 2022.

[20] T. Zhang*, V. Kishore*, F. Wu*, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

A Appendix

A.1 Response Generators

Response Generator	**Definition**
AMBIGUOUS_ASR_RESPONDER	triggered if the ASR confidence score for the user utterance is < 0.4
CHITCHAT_RESPONDER	responds to general chitchat by the user
DEFAULT_RESPONDER	the default responder when no responders are able to satisfy the user query, for example - "play music", "subject question", "change task"
DIY_STEP_DETAILS_RESPONDER	responder for when user asks for more details for a DIY task
INTRO_RESPONDER	generates responses to questions like "Who are you?" or "What can you do?"
LAUNCH_RESPONDER	triggered during bot launch to speak the opening phrase
NO_RESUME_RESPONDER	used if the user does not want to continue a previously ongoing task
OFFENSE_RESPONDER	responds if user utterance classified to be offensive
QA_RESPONDER	answers in-task and open-domain questions asked by the user using context
REPLACEMENT_RESPONDER	suggests replacement ingredients if user does not have the required ingredient
SHOW_STEPS_RESPONDER	responsible for all steps within a task, showing ingredients and displaying the recommended next tasks on finishing the current task
SUGGEST_RECIPE_RESPONDER	suggests recipes based on occasion, cuisine or any other parameter
TASK_COMPLETE_RESPONDER	farewell message to the user on exiting out of the taskbot
TASK_QUERY_RESPONDER	responsible for searching the corpus for a user search query
TASK_TIPS_RESPONDER	returns tips for an ongoing task when requested by the user
VIDEO_RESPONDER	responds with a video of the task if the task has one upon user request

Table 8: Response generators and their definitions

A.2 Intent Classification

Intent Class	Definition
search_recipe(name)	tries to find instructions about a specific recipe name
search_task	tries to find instructions about a specific task or DIY
suggest_recipe	asks for recipe recommendations with no specific recipe mentioned
answer	asks some question related to the current task or open-domain
replace(ingredient)	wants to find replacement for some ingredient or tool
next_step	wants to move to the next step
begin	explicitly says is ready to start working on the task/recipe
done	finished doing something
acknowledge	acknowledges the system's response
wait	tells the system to wait for some time
deny	is negative about the system's response
stop	wants to stop
show_materials	wants to see the list of required ingredients or tools or list of steps
overview	wants to see an overview of the task or recipe
more_results	wants to see more result options
repeat	wants the system to repeat what it said
goto_step(number)	wants to jump to some specific step
set_timer(duration)	wants to set a timer
change_task	wants to change the current task or recipe
diet_restriction(diet)	mentions some diet restriction he has
help	generic help request
thanks	thanks the system
chitchat	wants to talk about something random
offense	offends the system
nsfw	asks for something that is nsfw
illegal_action	wants to find info about some illegal action
dangerous_task	wants to do some task potentially dangerous
suicide_attempt	attempts to do something related to suicide
personal_information	mentions some of his personal information
not_understood	explicitly says he did not understand
very_specific	asks for an extremely specific task or DIY
subjective_qa	asks some very subjective question
command	various unsupported commands related to controlling devices or apps
product_search	wants to search for products such as books, etc.
play_song	want to play some given song
needs_clarification	when the user's request is unclear (eg. bad syntax, incomplete sentence)

Table 9: Intent classes and their definitions.

Intent	FlanT5-XL (fine-tuned)			GPT-4		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
acknowledge	0.7	0.8	0.74	0.3	0.85	0.45
answer	0.74	0.7	0.72	0.61	0.85	0.71
begin	0.78	0.9	0.84	0.74	1	0.85
chitchat	0.85	0.28	0.42	0.89	0.5	0.64
dangerous_task	0.67	0.8	0.73	0.83	0.25	0.38
deny	0.69	0.9	0.78	0.76	0.95	0.84
diet_restriction	0.75	0.75	0.75	0.95	0.9	0.92
done	0.66	0.96	0.78	0.88	0.79	0.83
financial_advice	0.74	0.85	0.79	0.86	0.95	0.9
goto_step	0.95	0.9	0.92	1	0.7	0.82
help	0.91	0.71	0.8	0.88	0.75	0.81
legal_advice	0.88	0.7	0.78	1	0.25	0.4
more_results	0.78	0.7	0.74	1	0.7	0.82
next_step	0.87	0.65	0.74	0.68	0.75	0.71
not_understood	0.71	0.85	0.77	0.56	0.95	0.7
nsfw	0.94	0.8	0.86	0.95	0.9	0.92
offense	0.49	0.95	0.64	0.76	0.95	0.84
overview	0.8	0.6	0.69	0.72	0.65	0.68
personal_information	0.64	0.7	0.67	0.93	0.65	0.76
repeat	1	0.71	0.83	0.65	0.81	0.72
replace	0.91	0.5	0.65	0.91	1	0.95
search_task	0.55	0.85	0.67	0.67	0.93	0.78
set_timer	1	0.9	0.95	1	0.9	0.95
show_image	0.79	0.75	0.77	1	0.9	0.95
show_ingredients	0.66	0.95	0.78	0.84	0.8	0.82
show_video	1	0.7	0.82	1	1	1
stop	0.71	0.71	0.71	0.95	0.64	0.77
suggest_recipe	0.71	1	0.83	0.54	0.95	0.69
suicide_attempt	0.83	0.5	0.62	1	0.6	0.75
thanks	0.82	0.7	0.76	0.76	0.8	0.78

Table 10: Intent classification results per class for FlanT5-XL (fine-tuned) and GPT-4. Some intents from the final taxonomy are not included here.

A.3 Image Retrieval

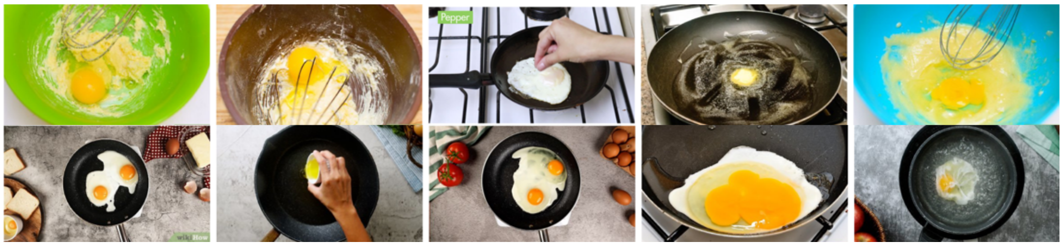


Figure 12: The top 5 retrieved images for the query 'Crack an egg into the skillet' using zero-shot CLIP (top) and fine-tuned (bottom).

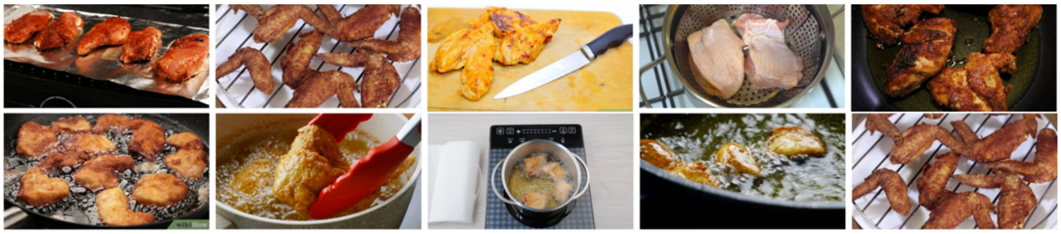
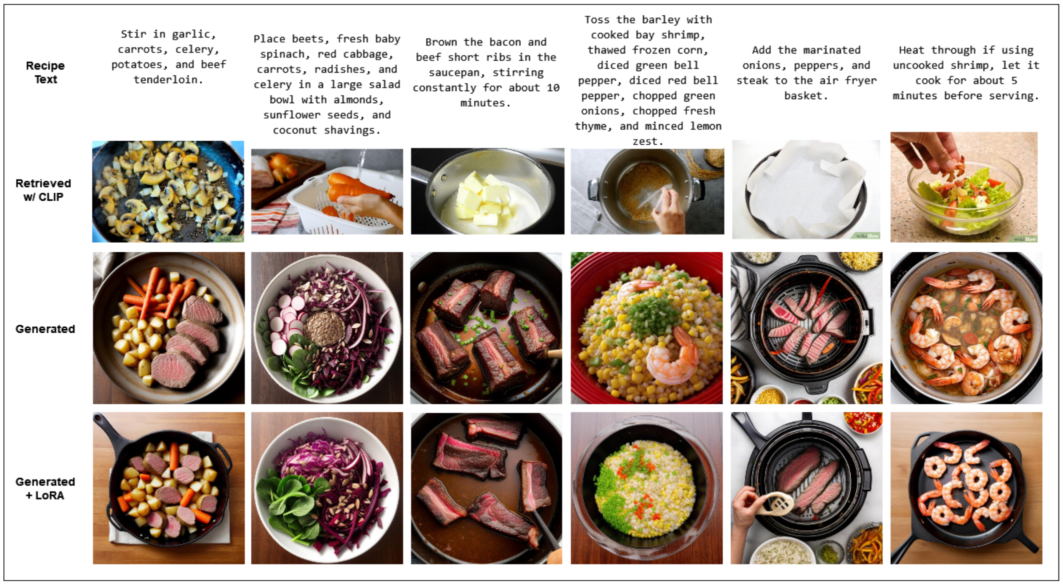


Figure 13: The top 5 retrieved images for the query 'Fry half of the wings at a time for about 5-7 minutes, until light golden' using zero-shot CLIP (top) and fine-tuned (bottom).

A.4 Generative Image Augmentation



Stir in garlic, carrots, celery, potatoes, and beef tenderloin.

Place beets, fresh baby spinach, red cabbage, carrots, radishes, and celery in a large salad bowl with almonds, sunflower seeds, and coconut shavings.

Brown the bacon and beef short ribs in the saucepan, stirring constantly for about 10 minutes.

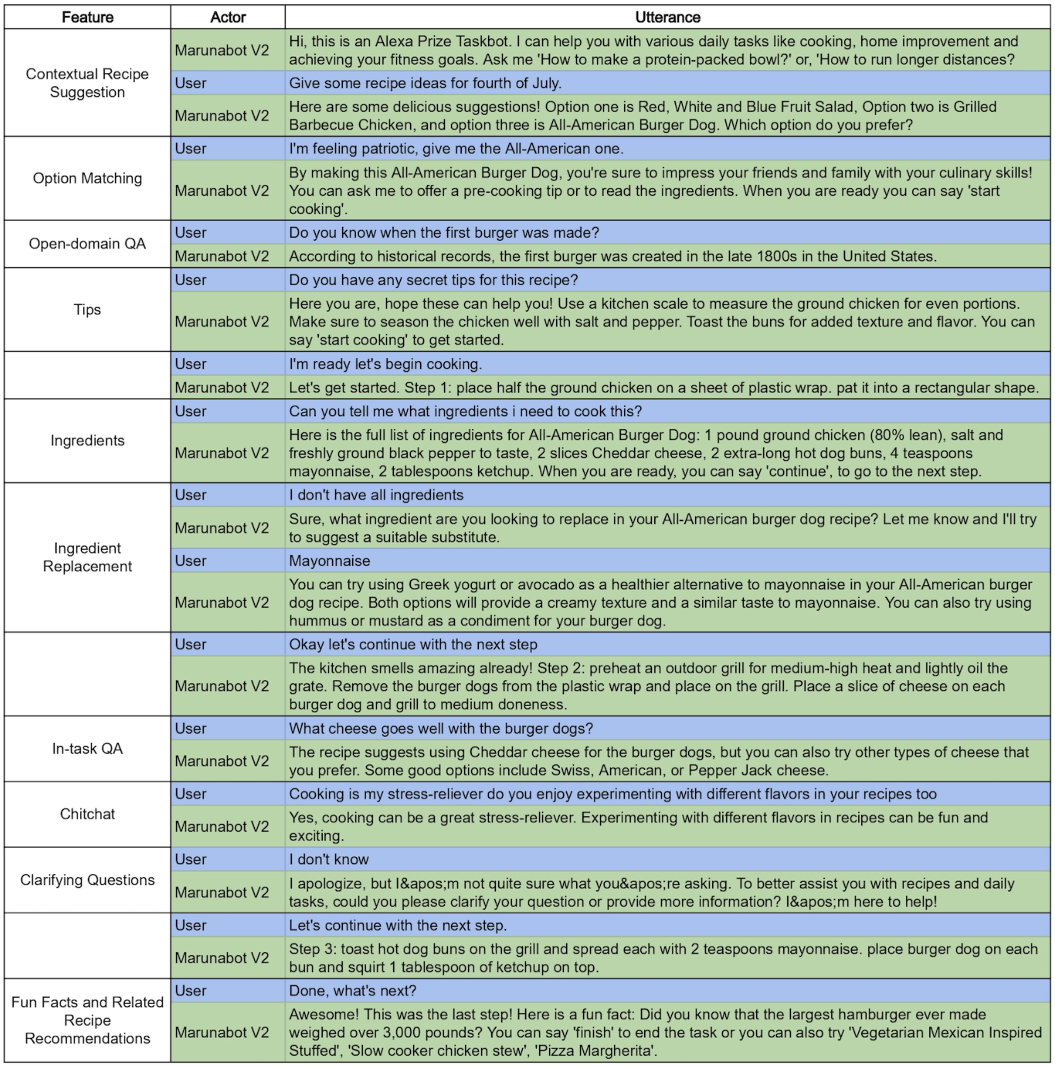
Toss the barley with cooked bay shrimp, thawed frozen corn, diced green bell pepper, diced red bell pepper, chopped green onions, chopped fresh thyme, and minced lemon zest.

Add the marinated onions, peppers, and steak to the air fryer basket.

Heat through if using uncooked shrimp, let it cook for about 5 minutes before serving.

Figure 14: Comparison between retrieved and generated images, with and without LoRA.

A.5 Full Conversation with Marunabot v2



Feature	Actor	Utterance
Contextual Recipe Suggestion	Marunabot V2	Hi, this is an Alexa Prize Taskbot. I can help you with various daily tasks like cooking, home improvement and achieving your fitness goals. Ask me 'How to make a protein-packed bowl?' or, 'How to run longer distances?
	User	Give some recipe ideas for fourth of July.
	Marunabot V2	Here are some delicious suggestions! Option one is Red, White and Blue Fruit Salad, Option two is Grilled Barbecue Chicken, and option three is All-American Burger Dog. Which option do you prefer?
Option Matching	User	I'm feeling patriotic, give me the All-American one.
	Marunabot V2	By making this All-American Burger Dog, you're sure to impress your friends and family with your culinary skills! You can ask me to offer a pre-cooking tip or to read the ingredients. When you are ready you can say 'start cooking'.
Open-domain QA	User	Do you know when the first burger was made?
	Marunabot V2	According to historical records, the first burger was created in the late 1800s in the United States.
Tips	User	Do you have any secret tips for this recipe?
	Marunabot V2	Here you are, hope these can help you! Use a kitchen scale to measure the ground chicken for even portions. Make sure to season the chicken well with salt and pepper. Toast the buns for added texture and flavor. You can say 'start cooking' to get started.
	User	I'm ready let's begin cooking.
	Marunabot V2	Let's get started. Step 1: place half the ground chicken on a sheet of plastic wrap. pat it into a rectangular shape.
Ingredients	User	Can you tell me what ingredients i need to cook this?
	Marunabot V2	Here is the full list of ingredients for All-American Burger Dog: 1 pound ground chicken (80% lean), salt and freshly ground black pepper to taste, 2 slices Cheddar cheese, 2 extra-long hot dog buns, 4 teaspoons mayonnaise, 2 tablespoons ketchup. When you are ready, you can say 'continue', to go to the next step.
Ingredient Replacement	User	I don't have all ingredients
	Marunabot V2	Sure, what ingredient are you looking to replace in your All-American burger dog recipe? Let me know and I'll try to suggest a suitable substitute.
	User	Mayonnaise
	Marunabot V2	You can try using Greek yogurt or avocado as a healthier alternative to mayonnaise in your All-American burger dog recipe. Both options will provide a creamy texture and a similar taste to mayonnaise. You can also try using hummus or mustard as a condiment for your burger dog.
	User	Okay let's continue with the next step
	Marunabot V2	The kitchen smells amazing already! Step 2: preheat an outdoor grill for medium-high heat and lightly oil the grate. Remove the burger dogs from the plastic wrap and place on the grill. Place a slice of cheese on each burger dog and grill to medium doneness.
In-task QA	User	What cheese goes well with the burger dogs?
	Marunabot V2	The recipe suggests using Cheddar cheese for the burger dogs, but you can also try other types of cheese that you prefer. Some good options include Swiss, American, or Pepper Jack cheese.
Chitchat	User	Cooking is my stress-reliever do you enjoy experimenting with different flavors in your recipes too
	Marunabot V2	Yes, cooking can be a great stress-reliever. Experimenting with different flavors in recipes can be fun and exciting.
Clarifying Questions	User	I don't know
	Marunabot V2	I apologize, but I&apos;m not quite sure what you&apos;re asking. To better assist you with recipes and daily tasks, could you please clarify your question or provide more information? I&apos;m here to help!
	User	Let's continue with the next step.
	Marunabot V2	Step 3: toast hot dog buns on the grill and spread each with 2 teaspoons mayonnaise. place burger dog on each bun and squirt 1 tablespoon of ketchup on top.
Fun Facts and Related Recipe Recommendations	User	Done, what's next?
	Marunabot V2	Awesome! This was the last step! Here is a fun fact: Did you know that the largest hamburger ever made weighed over 3,000 pounds? You can say 'finish' to end the task or you can also try 'Vegetarian Mexican Inspired Stuffed', 'Slow cooker chicken stew', 'Pizza Margherita'.

Figure 15: A complete conversation (not from real user) showcasing different features of the bot.