TWIZ-v2: The Wizard of Multimodal Conversational-Stimulus

Rafael Ferreira, Diogo Tavares, Diogo Silva, Rodrigo Valério, João Bordalo,
Inês Simões, Vasco Ramos, David Semedo, Joao Magalhaes
NOVA University of Lisbon, NOVA LINCS
{rah.ferreira,dc.tavares,dmgc.silva,r.valerio
j.bordalo,ir.simoes,vcc.ramos}@campus.fct.unl.pt
{df.semedo,jm.magalhaes}@fct.unl.pt

Abstract

In this report, we describe the vision, challenges, and scientific contributions of the Task Wizard team, TWIZ, in the Alexa Prize TaskBot Challenge 2022 [1]. Our vision, is to build TWIZ bot as an *helpful*, *multimodal*, *knowledgeable*, and *engaging* assistant that can guide users towards the successful completion of complex manual tasks. To achieve this, we focus our efforts on three main research questions: (1) Humanly-Shaped Conversations, by providing information in a knowledgeable way; (2) Multimodal Stimulus, making use of various modalities including voice, images, and videos; and (3) Zero-shot Conversational Flows, to improve the robustness of the interaction to unseen scenarios. TWIZ is an assistant capable of supporting a wide range of tasks, with several innovative features such as creative cooking, video navigation through voice, and the robust TWIZ-LLM, a Large Language Model trained for dialoguing about complex manual tasks. Given ratings and feedback provided by users, we observed that TWIZ bot is an effective and robust system, capable of guiding users through tasks while providing several multimodal stimuli.

1 Introduction

Helping users in real-world manual tasks is a complex and challenging paradigm [11, 6, 1], where it is necessary to leverage multiple information sources, provide several multimodal stimuli, and be able to correctly ground the conversation in a helpful and robust manner. In this work, we build upon the success of TWIZ [10] and introduce new features along with expanding existing ones. This results in an assistant capable of guiding a user through a task while keeping it engaging and stimulating.

With the aim of advancing Multimodal Conversational AI, we explore three main research questions encompassing several contributions:

- **RQ1: Humanly-Shaped Conversations** - Conversations should be fun and rewarding, yet reach a successful completion. To achieve this, we take an all-encompassing approach and propose novel ways of finding/creating a task, getting task highlights in the overview, and handling the task execution dialogue with TWIZ-LLM, a Large Language Model (LLM) trained specifically for supporting a robust interaction in the TaskBot domain.
- **RQ2: Multimodal Conversational Stimulus** - Interactions between a user and an assistant should make use of various stimuli to keep the conversation engaging. To this end, we leverage both text and visual content. Particularly, we expand the curiosities paradigm by generating more fun and contextual curiosities. On the visual side, we explore several image-generation methods and propose ways to make them more consistent with the target

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

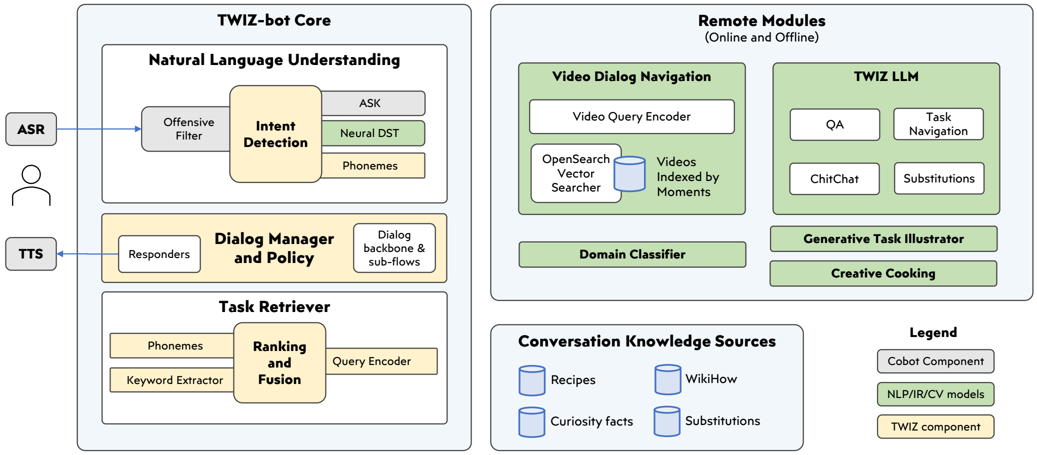

Figure 1: Architecture diagram of the TWIZ bot.

task. Finally, we present a novel Video Navigation feature, to allow for interactive video navigation by voice commands.

- **RQ3: Zero-shot Conversational Flows** - Given that users are unpredictable and may not follow the expected interaction, we need a robust dialogue framework. The goal is to steer the user through a pleasant and natural conversation while supporting these conversation detours. For this, we leveraged various LLMs and prompting strategies to extend TWIZ-LLM to converse about unseen topics with answers that are both in-scope and meaningful to the conversation.

2 TWIZ Bot Modular Architecture

An overview of our architecture is presented in Figure 1. It is built upon Amazon's CoBot [17], a framework made available by Amazon for developing conversational agents. The agent operates within AWS Lambda, while all machine learning algorithms run on external modules, resulting in a highly efficient architecture with optimal performance. Leveraging the framework's layered architecture, we adopt a shared database pattern using Amazon's DynamoDB.

2.1 Intent Detection

Understanding the user's intent is crucial to keeping a smooth flow in the dialogue interactions. Given its success, we use an approach similar to the previous year [10] by combining three methods: 1) phonemes-based matching; 2) rule-based corrections; and 3) a BERT-based model [34] for intent detection. These models work in an in-domain setting. However, they struggle to handle unseen or long-tail intents, requiring re-training to introduce new intents. To tackle this, we introduce a zero-shot intent detection method (Section 5).

2.2 Dialogue Manager

The dialogue manager keeps track of the current state and flow of the conversation. Our dialogue manager, adopts an event-driven state machine pattern. The progression through its different states is triggered by events associated with the detected intent of each user utterance, allowing for the context of a state to be used in order to provide relevant responses. With it, it is possible to keep track of the conversation's progress and leverage state transitions to maintain a correct conversation flow.

2.2.1 Dialogue Backbone-Flow

The dialogue manager provides graceful guidance to the user through the task at hand, by means of a *backbone-flow*, which is a set of states that are necessary to go through in order to progress through a task. In TWIZ, the backbone states are as follows:

1. Greeting - The starting state, in which TWIZ's functionalities are presented and task suggestions are made (e.g., Summer suggestions).

2. Grounding - State in which the selected suggestions or search results (Section 3.1) are shown to the user, so they can make a choice.

3. Task Overview - State after the selection of a particular task, where its overview – e.g. rating, duration, and others – is presented. This enables the user to either initiate the task or return to grounding in order to alter their selection. We also add a generated description to provide a brief task summary and further entice the user to start the task (Section 3.3).

4. Task Execution - The user can browse through the different steps of the task. Multimodal devices provide visual depictions of the steps, presented as either original or generated images (Section 4.2), or videos that the user can interact with through touch and voice (Section 4.3).

5. Task Completed - This is the last state of the dialogue, after the execution of the task. The agent offers more task suggestions, and the user has the option to either select one of these tasks, conduct a new search, or conclude the interaction.

2.2.2 Dialogue Sub-flows

Subsequently, the dialogue *backbone-flow* is enriched with *sub-flows*, comprised of a set of states associated with an additional feature, such as answering questions and engaging in chit-chat (Section 3.4), sharing curiosities (Section 4.4), and navigating through a video (Section 4.2).

The defined sub-flows are organized in a self-contained fashion, allowing for easy insertion, modification, or removal. The possibility to seamlessly integrate or remove sub-flows or states without causing disruptions to the rest of the state machine allows different developers to focus on various features without having to worry about conflicts with states, associated events, and response generators. The usage of different sub-flows also provides support for dialogue flexibility, allowing the user to stray from the main conversation in a contained way. By keeping track of the conversation's current state, the dialogue manager employs a stack-like checkpoint mechanism to offer seamless fallback options, guiding the user back to the *backbone-flow*.

While the primary objective of the dialogue manager is to assist the user in carrying out a task, it also accommodates the user's ability to switch tasks midway and provides the option to pause and resume the task at a later time, preserving all the progress made in that particular task.

3 RQ1: Humanly-shaped Conversations

Making sure that users select and successfully complete a task is one of the main objectives of a TaskBot. Therefore, the research goal is to deliver a conversation flow in an open, yet resilient manner. Our quest is to strike the delicate balance between the *task-knowledge* and a *humanly-shaped dialog* to ground the conversation on a complex manual task and guide the user through it.

3.1 Frictionless Task Suggestions and Search

TWIZ needs to swiftly ground the conversation in a frictionless way, by effectively and collaboratively finding the right task while also allowing for an exploration paradigm. Consequently, on the home page, we present a range of demonstrative tasks for users to select. These examples not only provide a clear overview of TWIZ's capabilities, but also offer users insights into the type of supported tasks. Moreover, from our analysis, a large portion of the time, users follow the suggestions provided, highlighting the value of these examples in guiding the users. Expanding upon these insights, we incorporated time-sensitive suggestions, to further improve the relevance of the tasks shown, as well as seasonal tasks relevant to the current time of the year. All the tasks suggested on the home page were manually selected to ensure a high-quality user experience.

The task search pipeline follows a multi-query, multi-ranking, and rank fusion approach. First, user queries are processed and the most relevant terms are extracted [10]. Secondly, we apply a ranking algorithm. Our approach involves carrying out both lexical (text) and semantic (embedding) searches on an OpenSearch index, with semantic searches conducted using embeddings generated by the MPNet model [31]. The results are then combined and re-ranked using the cosine similarity against the user query, and an additional step of heuristic-based task quality parameters, such as the presence of a video or the number of ratings. This two-step approach enhances our ability to deliver the most relevant and quality-assured tasks.

3.2 Creative Cooking: *What's in your fridge?*

With regards to cooking, a helpful TaskBot should be able to guide users through cooking deadlocks (e.g. missing an ingredient, no matching recipe). To this end, we introduce a novel *Creative Cooking* feature that seeks to push the limits of users' creativity, TWIZ's knowledge, and cooking. In particular, it seeks to boost the users' creativity by letting them create their own unique and personalized tasks, adjusted to their preferences and available ingredients.

For example, if the user has in the fridge a particular set of ingredients (e.g. zucchini and eggplant) and a particular cooking style in mind (e.g. vegetarian) the user should be able to prompt TWIZ with a recipe that has these particular constraints, resulting in a recipe such as "grilled veggies". To do this, we first extract ingredients and cooking styles from the user utterances, using a rule-based approach against a set of curated tags. This way, we can guarantee that only valid tags are used. After this, we provide the user with a set of recipes from the API that satisfies the user's requests, and we add a recipe generated using an LLM considering the user's specifications. In particular, we use a Vicuna [4] model, which we prompt to generate a title, a list of ingredients, and a list of steps, given the aforementioned set of tags. An example of the creative cooking feature can be seen in this video[1]. In Figure 7, we show an example of the creative cooking feature with a generated recipe, illustrated with our image generation methods (Section 4.2).

The creative cooking feature seamlessly integrates with all other TWIZ features (e.g. curiosities and image generation pipeline). As future work, we aim to create a user study for comparing manually created to generated recipes.

3.3 Task Overview: Task Promoter

When analyzing our interactions, we noticed that there is a direct correlation between users who start a task and higher ratings (Section 7). With this in mind, we looked for new ways to entice users to start a task during the *Task Overview* phase. Consequently, in the previous year, this led to the development of a 3D visual illustrator of the recipe [10] and manual templates to highlight certain features of a recipe [10], which resulted in low diversity responses. This year, we developed a Task Promoter, whose purpose is to generate, for any given task, a short and appealing description that highlights the best it has to offer.

3.3.1 Evaluation

We conducted a human evaluation, considering the recipes domain, and manually assessed for 100 recipes the preferred description: a prompted Vicuna-7B [4] or a fine-tuned GPT-2 [26] based model which we call RePro (for model training details refer to Appendix B.2). Additionally, we ask annotators to identify non-sensical descriptions and/or comprising ingredient hallucinations. To help assess the hallucinations, we provide annotators with both the recipe name and the ingredients, as in Table 7. We collect two annotations for each pair of descriptions.

Model	Win %	Hallucination%	Broken %	# Params
RePro (GPT-2)	20	57	**10**	770 M
Vicuna-7B	**61**	**29**	11	7000 M

Table 1: Results of manual evaluation of the recipe promoter. Ties are not included in *Win %*.

The results in Table 1 show that Vicuna in a zero-shot scenario resulted in preferred generations over a purpose-built smaller model. Although both models present very few non-sensical generations, a key differentiator is the hallucination of ingredients in the recipe, where over half of the RePro

[1] https://www.youtube.com/playlist?list=PLC5saXed4eNtMDJPITQOM4iOSGD83k_gy

descriptions had at least one hallucinated ingredient. However, we noticed that almost all hallucinated and broken descriptions are easily identified, allowing them to be automatically removed.

Given the success of the Vicuna-7B approach, we expanded this feature to the WikiHow domain in a similar way using the title of the task. To improve model performance, as a next step, we plan to train a *RePro* model based on outputs from an LLM such as GPT-4.

3.4 Task Execution: TWIZ LLM

One of our team's major focuses this year is to make TWIZ more natural and robust to user dialogues. Due to the emergence of effective LLM-based chatbots, users have higher expectations when interacting with TWIZ. To meet these expectations, we developed a task-oriented LLM-based approach, that seeks to support the *Task Execution* phase[2]. We focused only on recipes, but plan to further expand it to DIY tasks.

3.4.1 Conversational Data Augmentation

During the task execution phase, we want to allow users to fully explore the recipe rather than just advance to the next step, to be able to ask questions about the cooking process, replace an ingredient, and get advice on the tricky parts. To generate data that accounts for all of these sub-flows in a robust manner, we devised a dialogue-generation pipeline that leverages data gathered during our participation in the first edition of the Alexa TaskBot Challenge [11]. We complement this data with several data augmentation techniques, described below, to increase the diversity of the generated dialogues.

Conversation Flow The basis of the dataset is the policy that dictates how dialogues are generated. For this, we extracted user patterns from user interactions and created a directed graph containing all the identified intents and the probabilities of the user transitioning between them, allowing us to accurately model real-user behavior when generating a new dialogue. The intents considered are classified using a combination of rule-based heuristics and a customized Transformer-based intent classifier [10]. The complete list of intents considered and their description can be seen in Table 2. To improve dialogue diversity, we manually increased the transition probabilities of less common intents such as Ingredient Replacement and Questions. While, by doing this, we are making the generated dialogues policy diverge from the real-user policy, we believe the added frequency of such intents can greatly improve the ability of models trained on this data to attend to this type of intents.

As this dataset focuses on task execution, we do not consider the task selection and search phases and simulate the generated dialogues starting when the user asks the assistant to start an already selected task. An example of a generated dialogue can be seen in Appendix Table 8.

Task Selection As, in each dialogue, the user focuses on completing a single recipe. We used Amazon's provided recipes dataset and extracted 1000 recipes that had between 5 and 10 steps, and a total word count of no more than 350. This helps ensure that the recipes are long enough for a meaningful dialogue, with opportunities to ask questions, while also avoiding overly long recipes that could lead to repetitive or noisy dialogues.

User Queries For the data to closely mimic real user behavior, we used authentic user requests. To achieve this, we collected all user utterances classified for each considered intent and their absolute frequency. To clean up classification errors, we manually reviewed each of the most common user requests for each intent and removed any utterances that did not match the intent. When generating a new dialogue turn, the user utterance is selected from the list of utterances for the current intent using a random weighted selection, where the weight of each utterance is its absolute frequency, ensuring that more common utterances are more likely to be selected. However, some intents are extremely contextual, meaning that their associated user requests can have vastly different answers based on the relevant recipe (e.g., questions regarding a specific recipe step or ingredient replacement). For these cases, we utilize zero-shot LLM prompting and template-based approaches to produce natural-sounding and context-relevant user utterances. More information about this can be found in Appendix C.1.

[2]When the user has already selected and started a task.

	Intent	Description	Example	# ex.
Navigational	Next Step	Go to the next step of the task.	Next Step.	169
	Repeat	Repeats the last system utterance.	Repeat that.	76
	Stop	Ends the interaction.	Stop	73
	Yes	Confirmation (acts as Next Step).	Sure.	52
	Previous Step	Go to the previous step of the task.	Previous Step.	21
	Resume	Repeats the current step.	Resume.	18
Question	Get Curiosities	Asks for a fun fact.	Can you tell me a fun fact?	13
	Ingredients Replacement	Asks for a replacement.	I do not have sugar.	9
	Definition Question	Definition type question	What is a spatula?	4
	Question	Task specific question.	How much salt do I need?	–
Other	Fallback	Intent is not recognized.	Find a restaurant near me.	2618
	Sensitive	Mentions a sensitive/dangerous topic.	How do you make a nuke?	235
	Chit-Chat	Chit-chat utterances.	How are you today?	131
	Search	Ask for another task.	How to change a tire?	108
	More Detail	Ask for more details about a step.	More details, please.	36

Table 2: Intents list, description, example (manually-crafted) utterances, and the count of unique utterances available.

System Responses The system responses need to be diverse but also accurate and contextual, w.r.t. the recipe and the user request. To achieve this, based on the user intent, we use templates, knowledge bases, and LLM-prompting:

- **Navigational** - For these cases the system response consists only of step text.
- **Definition Question** - We query a dictionary[3] for the appropriate meaning of a given concept.
- **General Question** - Given a recipe, generate QA-pairs by prompting OpenAI's *text-davinci-003*. This way both the questions and system answers are contextual.
- **Ingredient Replacement** - Template responses, filled in using a knowledge base[4] of valid ingredient replacements (e.g., "{Ingredient A} can be replaced with {Ingredient B} in this recipe").
- **Fun facts** - We used OpenAI's *text-davinci-003* to generate curiosities for each recipe step by providing the step text and a relevant Wikipedia paragraph[5] and prompting the model to generate a fun fact relevant to that step.

For all other cases, a template-based approach was used. The templates were written by hand with up to 5 examples per case. In these cases, to ensure inter-dialogue response diversity, the selected response was randomly sampled from all templates not used in the past 5 dialogue turns. Additionally, every template is written in several different tones of voice, to allow for more diversity amongst the generated dialogues.

Dialogue Generation Pipeline The complete generation pipeline follows the following three steps when generating a new dialogue turn:

1. **Determine User Intent** - Based on the previous turn and on the extracted policy, sample the next user intent.
2. **Retrieve User Utterance** - Based on the selected intent and current recipe step, retrieve a candidate user utterance.
3. **Produce System Response** - Based on the selected intent, user utterance, and recipe, select the appropriate system response.

These steps are repeated until the user reaches the last step of the recipe or a *Stop* intent is selected.

[3] `https://github.com/wordset/wordset-dictionary`
[4] `https://foodsubs.com/`
[5] using `https://neuml.github.io/txtai/`

3.4.2 Base LLM Models

We are currently using two models: **Vicuna-7B** [4], a LLaMA-based [36] model fine-tuned on conversational data, and **OPT-1.3B** [42], to understand how different model sizes and architectures behave on this task.

For the model input, there are four key pieces of information that the model needs to be able to attend to (training prompts are shown in Appendix C.2):

- **Tone of Voice:** The tone that should be used by the system during the dialogue. It can be neutral, somewhat polite, polite, or very polite. This allows training models with controllable tone of voice in their responses.
- **Recipe Text:** The recipe title, followed by its steps.
- **Current Step:** The recipe step that the user is currently on. This helps in navigational requests and and in keeping the answers grounded.
- **Dialogue context:** The previous t turns + the *current user request*. For our evaluation, we use a $t = 1$.

3.4.3 Experimental Setup

Model Details We trained Vicuna-7B and OPT-1.3B on the same data. Vicuna was trained for 1 epoch, whereas OPT was trained for 10. In addition to SFT training, we also trained OPT under an RLHF paradigm [32, 23], to understand if the models would benefit from this approach.

Dataset We generated 10k dialogues with an 80/10/10 split. To create the negative system responses for RLHF, we employ different methods based on the user's intent:

- **Navigational** - we introduce a sentence from the previous/next step to simulate model copying mistakes.
- **Definition Questions** - we select the definition ranked lowest using a bag-of-words approach.
- **Ingredient Replacement** - we randomly choose an ingredient from a list to replace the current one.
- **Get a Curiosity** - we retrieve a curiosity with lower similarity using a bag-of-words approach.
- **Sensitive** - provides an answer to a sensitive request given by an uncensored LLM.

If no change is applied, and for all other intents, we introduce response perturbation and grammatical errors.

3.4.4 Automatic Evaluation

	Test Set %	OPT-SFT		OPT-RL		Vicuna	
		METEOR	BScore-F1	METEOR	BScore-F1	METEOR	BScore-F1
Navigational	65.36%	64.39	81.73	59.79	79.90	**98.36**	**99.29**
Get Curiosities	4.83%	14.94	57.44	14.52	57.11	**18.29**	**61.96**
Ingredients Replacement	1.11%	77.37	87.59	32.92	68.84	**79.91**	**89.19**
Definition Question	4.07%	**83.76**	**91.52**	83.09	91.23	81.25	91.23
Question	9.41%	54.01	78.52	53.26	78.20	**61.80**	**82.92**
Sensitive	4.07%	81.75	90.83	**91.00**	**95.30**	68.30	85.52
Other*	11.15%	68.97	83.57	42.66	73.06	**84.78**	**92.65**

Table 3: Per intent results for all three trained models. *Sensitive intent was separated from the Other intent group due to outlier results.

To measure model performance, we evaluate the model on all turns of the test set dialogues. For automatic metrics, we considered METEOR [3] and BERT-Score-F1 [43] to measure textual overlap and semantic similarity, respectively. In Table 3, we provide a detailed analysis of the models' performance, by separating turns with the "more static" intents, which require copying from the task information (e.g., *Navigational*) or returning a default response (e.g., *Other*), from the turns that require access to some external knowledge.

These early results show that the Vicuna-based model is capable of handling *Navigational* intents. It also outperforms both OPT-based models for almost all intents. For Vicuna, the outliers are the *Curiosity* requests. These answers are difficult to evaluate using automatic metrics since there can be various correct answers given a user's request. RL on the OPT model seems to have little impact in most cases. However, it does improve *Sensitive* requests, while having a negative impact on both *Ingredient Replacement* and on the *Other* intent group, indicating that the model might confuse these requests as dangerous tasks. We also noted that when handling sensitive requests, both OPT and Vicuna recognized the nature of the request and did not respond inappropriately.

3.4.5 Human Evaluation

To more accurately measure the models' performance, we conducted a human evaluation for Ingredient Replacement, Questions, and Curiosity user requests. We focused on these request types due to their reliance on external knowledge and the possibility of multiple correct answers. For the 3 cases, we selected 50, 100, and 30 examples from the test set, respectively. The annotators were asked to rate each one on a scale of 0 (wrong/irrelevant) to 2 (accurate/very relevant). For both ingredient replacement and questions, the criteria used was accuracy, whereas for curiosity requests, the relevancy of the generated curiosity w.r.t. the current recipe step was annotated. Annotators were instructed to rate 0 in cases where the generated utterance is incoherent or a clear hallucination.

Table 4 shows the results of the human evaluation. Vicuna outperforms both OPT models. Using RL on OPT had a negative impact on model performance on two tasks, and improved for *Question* requests. For *Curiosity* requests, OPT produces fairly relevant curiosities, on par with Vicuna, but performance worsens with the RL approach.

Model	**OPT-SFT**	**OPT-RL**	**Vicuna**
Question	1.30	1.48	**1.77**
Ingredient Replacement	0.84	0.76	**1.48**
Get Curiosities	1.23	0.90	**1.30**

Table 4: Human evaluation results of TWIZ LLM models on the Question, Ingredient Replacement, and Fun Fact intents, on a 0 to 2 scale.

In a later analysis of the reward model used with OPT-RL, we noted that the reward model did not learn to model the preferences appropriately. We believe this is due to the low quality of the preference data. However, these results, along with the results observed in automatic evaluation, show that RLHF has the potential to provide meaningful performance improvements but lacks consistency. Thus, future work should focus on improving the quality of the generated preference data, for example, using more diverse LLM prompting methods.

To conclude, TWIZ-LLM presents a step forward in having a model capable of guiding a user through the execution of a task while providing natural and helpful responses.

4 RQ2: Multimodal Conversation-Stimulus

Conversation stimulus is what advances a conversation, i.e., the user's desire to obtain the final outcome of the task and the road that leads to it. TWIZ provides a number of linguistic, visual, and cognitive stimuli to keep conversations natural and engaging.

4.1 UI and APL Templates

The user interface significantly influences the perceptions and experiences of the users. As such, the quality and functionality of the interface are crucial. To address this, we developed a new user interface that is more capable of showcasing TWIZ's features.

Our design philosophy leans towards minimalism, emphasizing a clean theme to promote clarity and ease of use. This approach encourages a seamless user experience, contributing to more efficient

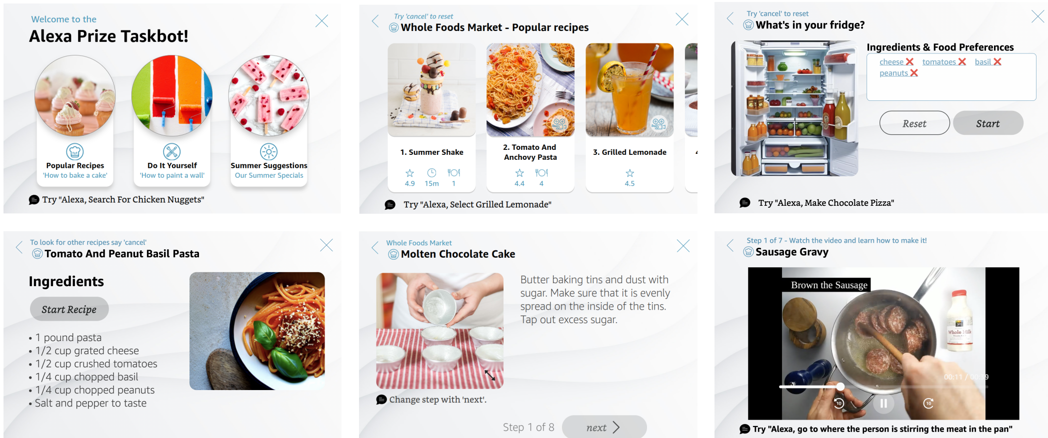

Figure 2: TWIZ's APL screens: *welcome screen*, *search results*, *"What's in my fridge?"*, *ingredients list*, *task step*, and *video dialogue*.

interactions. Additionally, on each screen, we provide a rotating set of contextual tips that introduce users to TWIZ's wide range of features. The complete set of screens can be found in Figure 2.

4.2 Task Visual Enrichment and Dialogue

Task illustrations are fundamental for rich and engaging visual-stimuli interactions. Nonetheless, a significant number of tasks either lack accompanying illustrations or have low-quality ones. Moreover, creative cooking (Section 3.2), offers the possibility to compose a recipe from scratch, requiring a strategy to illustrate it. In this section, we describe how image generation models [29], can provide a solution to generate task and step-specific illustrations.

4.2.1 Generative Task Illustration

Illustrating a task with text-to-image algorithms can be challenging, and requires extra care to ensure the generated images are grounded on the task and have good quality. Our implementation can be divided into three main steps:

1. **Image generation** is a non-trivial task, as it involves generating images from a task that might not be visual in nature. We propose a new method [37], which uses an LLM to convert the task into a visual prompt with high consistency that can then be used in text-to-image models. We generate multiple images for each task using the Stable Diffusion XL model [24].
2. The **image score** is given by an alignment score. This alignment score is calculated by the NL2VI metric [37], which measures the alignment of the image to the prompt. The alignment is calculated using VQA algorithms [20, 19, 41, 40]. The questions for VQA are generated using an LLM with in-context learning and are filtered using question answering [16] and NLI [22] models. This score ensures that the image is grounded on the task. The results of Table 5 show that our method achieves better results than the TIFA [14] baseline.
3. Lastly, we **rank the generated images**, as well as the original image, according to their score and choose the one with the highest score to illustrate the task [37].

4.2.2 Generative Recipe Step Illustration

After having a method to illustrate the main tasks' content, we now want to illustrate every step of a task using an image-generation approach, with visual continuity guarantees, which we refer to as *Coherent Step Illustration*. As established in Section 4.2.1, generated images must be consistent with the prompts, ensuring the desired elements are present in the image. This is important in task step illustration, as the illustrations must be faithful to the step to steer the user in the right direction.

	Equality	NLI	BERT-Score
TIFA [14]			
Recipes	64.7	68.9	72.0
WikiHow	56.1	61.6	64.2
NL2VI [37]			
Recipes	73.7	76.2	**79.8**
WikiHow	73.1	**75.8**	73.8

Table 5: Comparison of the image alignment score between our method NL2VI [37] and the previous SOTA TIFA [14].

The main focus of this method is the sequential nature of step illustrations. Unlike *Task Illustration*, which is independent, *Step Illustrations* have dependencies regarding the previous steps.

Our method takes as input a sequence of steps $\{S_1, S_2, ..., S_n\}$ and generates each image, I_n, which illustrates step S_n by considering all previous steps. The motivation is to increase the coherence of the sequence of images. The method comprises three phases:

1. **Sequence of Captions.** When generating an illustration for a recipe step, we want the illustration of step S_n to consider the previously generated images more directly. Hence, to address this limitation, we propose to consider the captions [8] of the previously generated images, instead of the text from previous steps. With these captions, our updated input can be written as, $\{C_1, C_2, ..., S_n\} \rightarrow I_n$.
2. **Sequence-to-Prompt LLM.** To transform the sequence $\{C_1, C_2, ..., S_n\}$ into an appropriate image generation prompt, we train an LLM with sequences of (captions+step) and the target prompt.
3. **Sequence-to-Prompt Generation.** After training, we can generate the prompts by iteratively prompting the LLM with the current step and the previous captions. These generated prompts contain enough information to increase the coherence of the generated images, with respect to the previous generations.

Given the nature of the task, we only apply this method to the recipes domain. In future work, we will conduct a user study to measure the performance of this approach.

4.3 Video Moment Retrieval

In order to facilitate the interaction with the vast amount of information present in a video, we developed a method to enable users to control videos using voice commands, when the video is in full-screen mode. This allows users to interact with videos effortlessly and intuitively, enhancing their overall viewing experience, as in Figure 4. An example interaction can be seen in this video[6].

As a first step, we extracted keyframes from all videos. For each keyframe, we generated a caption, using InstructBLIP [8]. Then we used the image and caption of each frame to extract embeddings using CLIP [25]. All of this information is stored in an OpenSearch index, along with the task metadata. The search is performed using the user query (e.g. "*When did the chef mix in the flour?*"), and uses both a full-text and embedding-based (text and image) search. The result thus has three sets of frames, one for each type of embedding (text and image) and one for plain text. We then perform re-ranking of the potential frames based on rank fusion. After

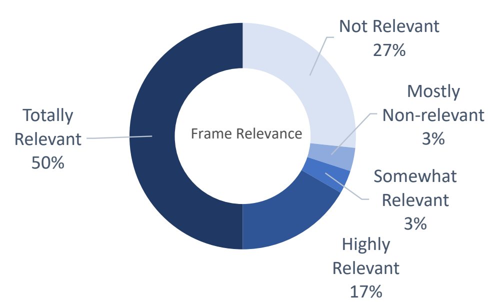

Figure 3: Video moment retrieval results.

[6]https://www.youtube.com/playlist?list=PLC5saXed4eNsebM8C4W5S_BQ9ADEgwH57

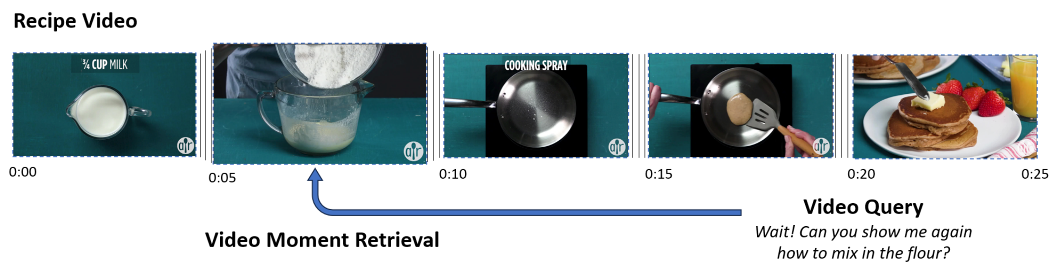

Figure 4: TWIZ users can ask a question about a video and navigate to the correct video moment that answers the question. An example interaction can be seen in this demo video link.

selecting the winning frame, we perform a seek in the video based on the timestamp of the frame on the top of the rank.

To evaluate the proposed method, we ran a test of 30 queries over 10 videos. For each user query, a video frame is returned. We judged the frame's relevance on a scale from 1 to 5, with 1 indicating a non-relevant frame and 5 indicating a totally relevant frame. As shown in Figure 3, 27% of the results were not relevant to the query, while 50% were considered totally relevant. Given these results, it is important to study ways to reduce the occurrences of lower-quality results by making better use of video information and improving the ranking algorithm.

4.4 Fact-Grounded Curiosity Generation

Curiosity-stimuli, i.e. *fun-facts*, turn conversations into an educational and memorable experience. In the first edition of the TaskBot Challenge, we contextually enriched dialogues with a manually curated set of curiosities [10]. Subsequently, we performed an A/B test with over 1000 conversations, which showed that curiosities increase user engagement and can also provide an average rating improvement [39].

Given these encouraging results, this year we expand this paradigm by using LLMs to generate more contextual and interesting curiosities, removing the bottleneck of manually creating curiosities. In particular, we use GPT-4 as the backbone for our curiosity generation model. Given that LLMs have a tendency to hallucinate [15], we add a relevant Wikipedia passage as context to ground the generation, alongside the task title and current step. We conduct a user study comparing the manually annotated curiosities with the generated ones, according to the following aspects: (1) **relevance** to the step, (2) **truthfulness** according to the provided information, and (3) **fun factor**. These aspects are measured on a scale of 0 (low) to 2 (high). Additionally, we also ask annotators to choose a winner between the two methods. In total, we collect 150 annotations with 3 annotations per comparison. These results are shown in Table 6.

	Win %	Relevance	Truthfulness	Fun Factor
Manually Curated	33.33%	0.71	1.78	0.80
Generated	66.67%	1.56	1.50	0.90

Table 6: Comparison between manually curated curiosities and curiosities generated by GPT-4.

The results show that the annotators have a clear preference for the generated curiosities, despite their truthfulness value being lower than the manually curated ones. Regarding the fun factor, none of the methods exhibits a high score. This suggests that the concept of "fun" in this domain requires further exploration. Nevertheless, the generated curiosities reduce the bottleneck of creating a manual set of curiosities and tend to be more fun and considerably more relevant.

5 RQ3: Zero-Shot Conversation Ramblings

Often, users elaborate their responses or ask for side information, which conventional systems, with a strict dialogue flow, fail to respond. TWIZ aims to improve the user experience by adapting *on-the-fly to conversation ramblings* introduced by the user, through zero-shot approaches.

Zero-shot DST as Reading Comprehension A rapidly evolving dialogue system such as TWIZ requires DST modules that easily support the inclusion of new slots and intents, even with little to no available data. To achieve this, we follow *Namazifar et al.* [21] to cast the typical zero-shot DST task as a reading comprehension one. To train our models, we require QA examples. These are automatically extracted from unlabeled TWIZ data, in a self-supervised setting [35]. Furthermore, during inference time, we require one question per slot and intent. These are created by prompting an LLM (*text-davinci-003)* to generate one question per slot. This contrasts with typical approaches, where questions are either derived from templates or manually written [21]. We find that the strategy of pre-training the model using in-domain dialogues and LLM-based questions significantly improves performance, as detailed in our work [35].

This model is used when new slots and intents are required to support new features, and while the team is gathering relevant, in-domain, data. When enough data is available, we pivot towards full-shot approaches (Section 2.1).

Zero-Shot Responses with TWIZ-LLM Generative Large Language Models trained with Human Feedback [23] possess a great ability to answer to a wide range of user ramblings. After the release of ChatGPT and its open-source competitors [4, 33], it became clear that LLMs wrapped in strong guard rails [23, 2] can respond to chit-chat, question-answering, and fallback intents in an appropriate and pleasant way. In this context, we use the responses given by TWIZ-LLM (Section 3.4) to respond to many of these requests.

6 Trustworthy TaskBot Generative Vision and Language

Given the recent advancements in using generative methods for both text [4, 23, 33] and image generation [30], a recent and very important research direction is how to guarantee trustworthy and consistent generations [13, 37]. In our work, we make use of various generative methods and we always ensure that they are as trustworthy as possible through the implementation of several guardrails and verification methods.

In text generation tasks that are grounded in some input constraint, such as the *Creative Cooking* that uses only valid tags for generating new recipes, the *Task Promoter*, and the *Curiosities* which receive a grounding Task/Wikipedia passage, are methods that can be further checked by a verification method such as *True* [12] and *Q2* [13], which use question generation/answering methods to ensure that the output is factually consistent with the input.

In the case of images, we take specific care in generating images that are *Consistent* with the prompt through the use of a novel verification pipeline NL2VI [37] and further expand this work to generate images that are *Coherent* along a sequence of steps. These methods are based on the creation of a *visual prompt* which is more suitable for image generation, followed by a T2I pipeline where images are verified through the use of VQA algorithms.

We believe that these methods are a step forward in diminishing the problems found in generative methods with the aim of providing more relevant and accurate information to the user.

7 User Interaction Analysis

User Behavior Understanding user's interactions allows us to discover possible interaction bottlenecks or new conversational paths.

Looking at device type, we see that on average, ratings from headless interactions are higher than multimodal ones 3.47 vs. 3.27, respectively. The use of multimodal devices also led to the increase of users using the on-screen buttons, rather than their voice, with over 35% of the interaction turns being taps on the screen. Another typical user behavior is asking for commands that our bot cannot

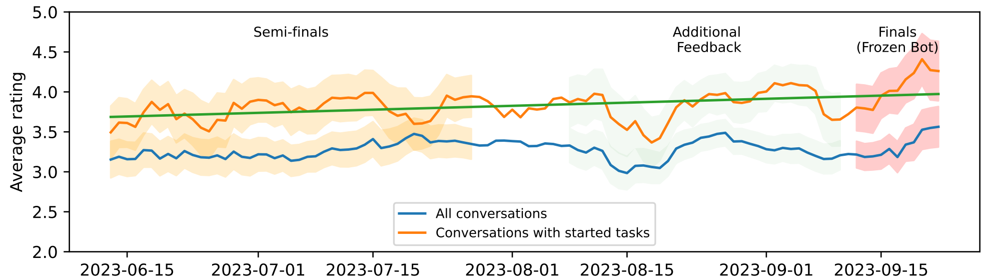

Figure 5: TWIZ's 7-Day average rating since the semi-finals for conversations with at least 3 turns.

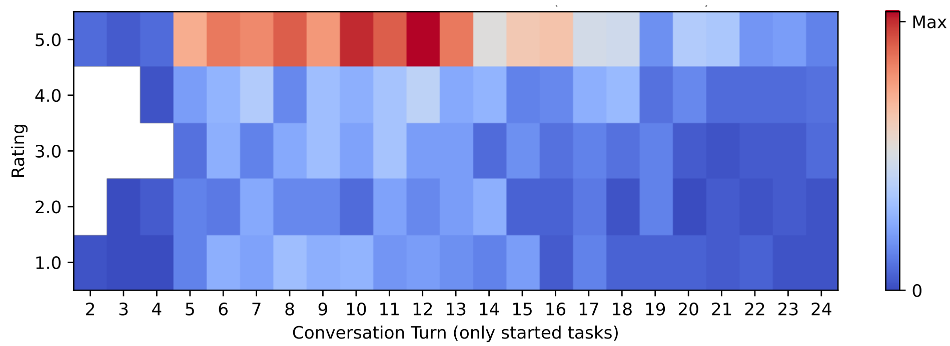

Figure 6: Conversations' ratings per turn when a task is started.

comply with, such as playing YouTube videos or music, which happens in 8.4% of interactions. We also see that non-rated conversations tend to be very short, with an average of 2.35 vs. 7.46 turns for rated ones. When focusing on ratings and task execution, we see that 32% of users who rate, start a task. From these, 42.8% start a recipe while the remaining 57.2% do a WikiHow task. Additionally, they also rate significantly higher than those who don't start (3.86 vs. 3.30, respectively). Looking at the task type, users who start a recipe give an average rating of 4.07 whereas users who start a WikiHow have an average rating of 3.70.

Ratings Progression In Figure 5, we present the average 7-Day rating since the semi-finals. There are two major observations: 1) *an increase in the user ratings throughout this period*, and 2) *TWIZ's high effectiveness in conversations in which users start a task*. Moreover, a positive trend in the ratings can be observed from the start of the semi-finals. In some periods, ratings variability is high, specially in the middle of each stage's period, which we primarily attribute to intense system modifications.

Ratings per Conversation Length Finally, Figure 6, depicts conversations rating per turn and provides further evidence of TWIZ bot's effectiveness in delivering the Alexa Prize TaskBot Challenge's main objective - guiding an user through a complex task. Namely, it can be seen that a large portion of conversations in which users start a task, get the maximum rating (top line). These results highlight TWIZ bot's consistency and quality during a comprehensive time frame and for users who are using the bot in its full capacity.

8 Conclusions and Future Work

In this paper, we summarized TWIZ's work on the second edition of the Alexa TaskBot Challenge. We built upon the strong foundation of the previous year and provided several new contributions:

- We focused on ensuring an engaging interaction by supporting *natural and knowledgeable* dialogues. To achieve this, we introduced the "Creative Cooking" feature, empowering users to craft their own custom recipes. Adding to this, we promote tasks in a positive way to increase user engagement, and provide navigational and contextual responses to user requests using TWIZ-LLM, an LLM trained specifically for the TaskBot domain.

- We provided various *multimodal stimuli* to the user. We expanded the curiosities paradigm and focused on improving multimodal interactions by generating task illustrations. In this last point, we proposed new methods to illustrate the task as well as its steps, with the added care to make illustrations consistent throughout the task. Still, in the multimodal stimuli, we also developed a video moment retrieval pipeline allowing video navigation by voice.
- Finally, we created a *robust* system to allow for more *flexible* user interactions and analyzed the users' interactions with the system resulting in several relevant insights.

In future work, we aim to continue exploring the use of LLMs and expanding their utilization to all stages of the dialogue, as well as introducing new multimodal stimuli, while assessing their impact on the conversation.

References

[1] 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*. `https://www.amazon.science/alexa-prize/proceedings/alexa-lets-work-together-introducing-the-second-alexa-prize-taskbot-challenge`

[2] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *CoRR* abs/2204.05862 (2022). `https://doi.org/10.48550/arXiv.2204.05862` arXiv:2204.05862

[3] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Association for Computational Linguistics, Ann Arbor, Michigan, 65–72. `https://aclanthology.org/W05-0909`

[4] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. `https://lmsys.org/blog/2023-03-30-vicuna/`

[5] Jason Ingyu Choi, Ali Ahmadvand, and Eugene Agichtein. 2019. Offline and Online Satisfaction Prediction in Open-Domain Conversational Systems. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. ACM, 1281–1290. `https://doi.org/10.1145/3357384.3358047`

[6] 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, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*. International Committee on Computational Linguistics, 3514–3529. `https://aclanthology.org/2022.coling-1.310`

[7] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free Dolly: Introducing the World's First

Truly Open Instruction-Tuned LLM. https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

[8] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. *CoRR* abs/2305.06500 (2023). https://doi.org/10.48550/arXiv.2305.06500 arXiv:2305.06500

[9] Rafael Ferreira, David Semedo, and João Magalhães. 2023. Rating Prediction in Conversational Task Assistants with Behavioral and Conversational-Flow Features. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*. ACM, 2314–2318. https://doi.org/10.1145/3539618.3592048

[10] Rafael Ferreira, Diogo Silva, Diogo Tavares, Frederico Vicente, Mariana Bonito, Gustavo Goncalves, Rui Margarido, Paula Figueiredo, Helder Rodrigues, David Semedo, and Joao Magalhaes. 2022. TWIZ: A conversational Task Wizard with multimodal curiosity-exploration. In *Alexa Prize TaskBot Challenge Proceedings*. https://www.amazon.science/alexa-prize/proceedings/twiz-a-conversational-task-wizard-with-multimodal-curiosity-exploration

[11] Anna Gottardi, Osman Ipek, Giuseppe Castellucci, Shui Hu, Lavina Vaz, Yao Lu, Anju Khatri, Anjali Chadha, Desheng Zhang, Sattvik Sahai, Prerna Dwivedi, Hangjie Shi, Lucy Hu, Andy Huang, Luke Dai, Bofei Yang, Varun Somani, Pankaj Rajan, Ron Rezac, Michael Johnston, Savanna Stiff, Leslie Ball, David Carmel, Yang Liu, Dilek Hakkani-Tur, Oleg Rokhlenko, Kate Bland, Eugene Agichtein, Reza Ghanadan, and Yoelle Maarek. 2022. Alexa, let's work together: Introducing the first Alexa Prize TaskBot Challenge on conversational task assistance. In *Alexa Prize TaskBot Challenge Proceedings*. https://www.amazon.science/publications/alexa-lets-work-together-introducing-the-first-alexa-prize-taskbot-challenge-on-conversational-task-assistance

[12] Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating Factual Consistency Evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*. Association for Computational Linguistics, 3905–3920. https://doi.org/10.18653/v1/2022.naacl-main.287

[13] Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q^2: Evaluating Factual Consistency in Knowledge-Grounded Dialogues via Question Generation and Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 7856–7870. https://doi.org/10.18653/v1/2021.emnlp-main.619

[14] Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A. Smith. 2023. TIFA: Accurate and Interpretable Text-to-Image Faithfulness Evaluation with Question Answering. *CoRR* abs/2303.11897 (2023). https://doi.org/10.48550/arXiv.2303.11897 arXiv:2303.11897

[15] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* 55, 12 (2023), 248:1–248:38. https://doi.org/10.1145/3571730

[16] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing Format Boundaries with a Single QA System. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 1896–1907. https://doi.org/10.18653/v1/2020.findings-emnlp.171

[17] Chandra Khatri, Anu Venkatesh, Behnam Hedayatnia, Raefer Gabriel, Ashwin Ram, and Rohit Prasad. 2018. Alexa Prize — State of the Art in Conversational AI. *AI Magazine* 39, 3 (Sep. 2018), 40–55. https://doi.org/10.1609/aimag.v39i3.2810

[18] Cat P. Le, Luke Dai, Michael Johnston, Yang Liu, Marilyn A. Walker, and Reza Ghanadan. 2023. Improving Open-Domain Dialogue Evaluation with a Causal Inference Model. *CoRR* abs/2301.13372 (2023). https://doi.org/10.48550/arXiv.2301.13372 arXiv:2301.13372

[19] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, He Chen, Guohai Xu, Zheng Cao, Ji Zhang, Songfang Huang, Fei Huang, Jingren Zhou, and Luo Si. 2022. mPLUG: Effective and Efficient Vision-Language Learning by Cross-modal Skip-connections. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Association for Computational Linguistics, 7241–7259. https://doi.org/10.18653/v1/2022.emnlp-main.488

[20] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 12888–12900. https://proceedings.mlr.press/v162/li22n.html

[21] Mahdi Namazifar, Alexandros Papangelis, Gökhan Tür, and Dilek Hakkani-Tür. 2021. Language Model is all You Need: Natural Language Understanding as Question Answering. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*. IEEE, 7803–7807. https://doi.org/10.1109/ICASSP39728.2021.9413810

[22] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 4885–4901. https://doi.org/10.18653/v1/2020.acl-main.441

[23] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*. http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html

[24] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *CoRR* abs/2307.01952 (2023). https://doi.org/10.48550/arXiv.2307.01952 arXiv:2307.01952

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 8748–8763. http://proceedings.mlr.press/v139/radford21a.html

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

[27] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *CoRR* abs/2305.18290 (2023). https://doi.org/10.48550/arXiv.2305.18290 arXiv:2305.18290

[28] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/v1/D19-1410

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 10674–10685. https://doi.org/10.1109/CVPR52688.2022.01042
[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 10674–10685. https://doi.org/10.1109/CVPR52688.2022.01042
[31] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. https://proceedings.neurips.cc/paper/2020/hash/c3a690be93aa602ee2dc0ccab5b7b67e-Abstract.html

[32] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize from human feedback. *CoRR* abs/2009.01325 (2020). arXiv:2009.01325 https://arxiv.org/abs/2009.01325

[33] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[34] Diogo Tavares, Pedro Azevedo, David Semedo, Ricardo Sousa, and Joao Magalhaes. 2023. Task Conditioned BERT for Joint Intent Detection and Slot-filling. In *Progress in Artificial Intelligence - 22nd EPIA Conference on Artificial Intelligence, EPIA 2023, Faial, Portugal, September 5 - 8, 2023, Proceedings*. Springer.

[35] Diogo Tavares, David Semedo, Alexander Rudnicky, and João Magalhães. 2023. Learning to Ask Questions for Zero-shot Dialogue State Tracking. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*. ACM, 2118–2122. https://doi.org/10.1145/3539618.3592010

[36] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023). https://doi.org/10.48550/arXiv.2302.13971 arXiv:2302.13971

[37] Rodrigo Valerio, Joao Bordalo, Michal Yarom, Yonatan Bitton, Idan Szpektor, and João Magalhães. 2023. Transferring Visual Attributes from Natural Language to Verified Image Generation. *CoRR* abs/2305.15026 (2023). https://doi.org/10.48550/arXiv.2305.15026 arXiv:2305.15026

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NeurIPS*. http://papers.nips.cc/paper/7181-attention-is-all-you-need

[39] Frederico Vicente, Rafael Ferreira, David Semedo, and Joao Magalhaes. 2023. The Wizard of Curiosities: Enriching Dialogues with Fun-facts. In *Proceedings of the 24rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, Prague, CZ. https://sigdialinlg2023.github.io/paper_sigdial75.html

[40] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. GIT: A Generative Image-to-text Transformer for Vision and Language. *Trans. Mach. Learn. Res.* 2022 (2022). https://openreview.net/forum?id=b4tMhpN0JC

[41] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 23318–23340. https://proceedings.mlr.press/v162/wang22al.html

[42] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. *CoRR* abs/2205.01068 (2022). https://doi.org/10.48550/arXiv.2205.01068 arXiv:2205.01068

[43] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=SkeHuCVFDr

A Creative Cooking: *What's in your fridge?* - Interaction

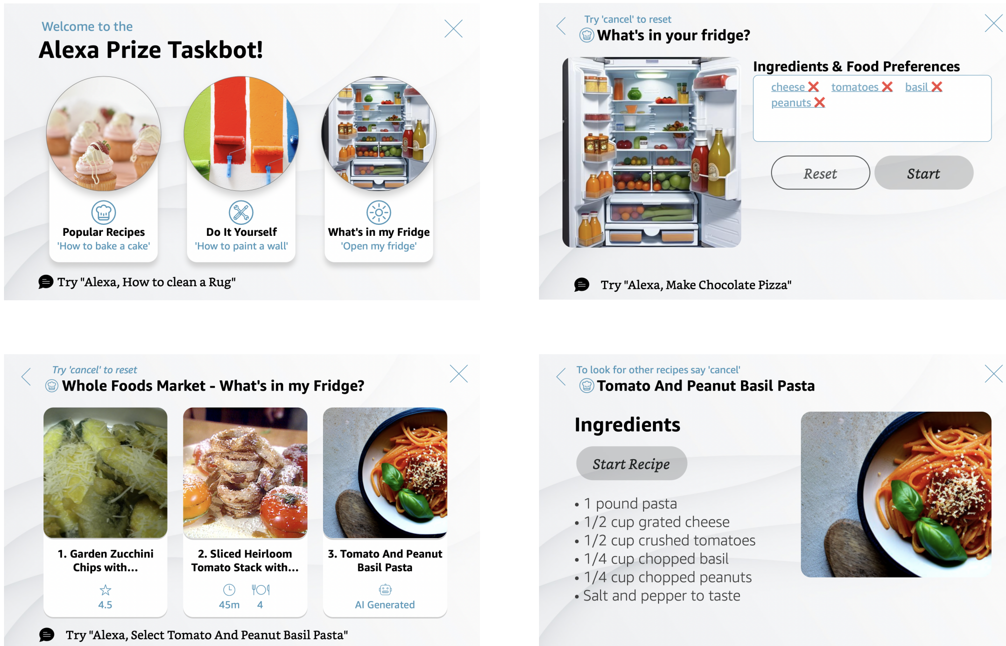

Figure 7: Creative cooking interaction examples.

B Task Promoter

B.1 Example Outputs

Recipe Title	Ingredients	Promoter Output
Lemony Lentil Salad	[cauliflower, green lentils, couscous, lemon, garlic, honey, extra virgin olive oil, parsley, feta cheese]	Lemony Lentil Salad: A zesty, plant-based meal that's packed with flavor and nutrients. Cauliflower, lentils, and couscous come together in a tangy dressing for a satisfying and healthy salad. Top with feta for a pop of flavor and crunch.
Garlic and Cheese Popovers	[butter, chives, lemon, egg, garlic, milk, flour, salt, ground pepper, parsley]	Upgrade your dinner game with these garlic and cheese popovers. Perfect for a cozy night in or a special occasion, they're sure to impress. Trust us, your taste buds will thank you!
Apple Crumble	[apple, sugar, lemon, ground cinnamon, nutmeg, ginger root, salt, flour, oats, dark brown sugar, ground cinnamon, butter]	This is my favorite homemade apple crumble recipe because it's simple to make but comes out perfectly fluffy and light!

Table 7: Promoter example outputs.

B.2 Training Details

To train the *RePro* model, we used 5k recipe descriptions along with their ingredients and steps. For the model, we used a GPT-2-774M [26], trained on instruction data [7]. We then further fine-tuned this model with Direct Preference Optimization (DPO) [27]. Given that DPO requires preference data, we prompted an LLM [4] for recipe descriptions and discarded any that were hallucinated (e.g. wrong number of ingredients mentioned, or wrong ingredients mentioned).

C TWIZ-LLM

C.1 Dataset

Task Title: Creamy Pesto Chicken Soup with Avocado		
Turn #	**User**	**System**
1	-	Step 1: Pour the Chicken Stock in a pan and add Garlic.
2	Got it done.	Step 2: Bring the stock to a boil over high heat and then add the Chicken Breast.
3	I do not have chicken breasts.	No problem, you can also use Turkey.
...	...	...
7	Tell me a fun fact.	In the US, on the 31st of July, it's national avocado day.
8	Next.	Step 8: Scoop the avocado chicken soup into bowls.
9	Next step.	We have reached the end of the task.
10	Stop.	Glad I could help you! See you again soon!

Table 8: Example conversation from TWIZ LLM dataset with manually-crafted utterances.

Questions. For the questions, we prompted OpenAI's API[7] to generate both question and answer, given the step text.

Ingredients Replacement - To simulate user utterances requesting an ingredient replacement, we apply a set of templates and fill in with an ingredient of a step. The system answer is also template-based, by accessing an external database of ingredient substitutes[8].

Definition Question - We extract noun phrases from the current step using spaCy and randomly pick one to fill in a "What is?" type question. The answer is extracted from a dictionary[9], using the intersection between the step's text and the various definitions to select the best one.

C.2 Example Prompts

Tables 9 and 10 show the prompts used to train the OPT and Vicuna models, respectively.

D Automated Testing

D.1 User Simulator

As TWIZ becomes increasingly more complex and undergoes constant changes, ensuring its correct behavior during testing becomes more challenging. To address this bottleneck in development, we created a testing tool that utilizes CoBot's interactive mode [17] to simulate user interactions. This user simulation can be conducted using either previously recorded user interactions or generated ones. By making the testing process automatic, the testing time can be shortened, which accelerates the R&D cycle.

While previously recorded user interactions help check if previous features weren't compromised while adding new ones, the generated user interactions allow to explore previously unseen conversation flows, allowing for the creation of a more robust assistant. In order to generate these interactions, our user simulation tool leverages the dialogue manager state-machine pattern implementation (Section 2.2) and user conversation transcripts. The transcripts provide the necessary data to compute

[7]specifically, *text-davinci-003*
[8]`https://foodsubs.com/`
[9]`https://github.com/wordset/wordset-dictionary`

```
Below is a recipe and the steps to make it, the step of the recipe that the user is
doing, and part of the dialogue between you and a user about this recipe. Write a
response that appropriately responds to the user request. Respond with a neutral
tone.

Recipe:
Garlicky Broccoli. Step 1: Wash Broccoli in water and shake off excess water.
Chop top (head) of the Broccoli into bite-sized pieces. Step 2: Heat Extra-Virgin
Olive Oil in a large nonstick skillet over medium-high heat. Step 3: Add Garlic to
the pan by crushing with a garlic press. (other steps omitted)

Current Step:
Step 1: Wash Broccoli in water and shake off excess water. Chop top (head) of the
Broccoli into bite-sized pieces.

Dialog:
User: Is it necessary to wash the broccoli before chopping? System: Yes, it is
important to wash the broccoli before chopping to remove any dirt or debris. User:
next.

Response:
Heat Extra-Virgin Olive Oil in a large nonstick skillet over medium-high heat.
```

Table 9: Example of the input prompt to the OPT model.

```
<|prompter|> I will give you a recipe and I want you to help me do it step by step.
Please use a neutral tone of voice. Recipe: Garlicky Broccoli. Step 1: Wash
Broccoli in water and shake off excess water. Chop top (head) of the Broccoli into
bite-sized pieces. Step 2: Heat Extra-Virgin Olive Oil in a large nonstick skillet
over medium-high heat. Step 3: Add Garlic to the pan by crushing with a garlic
press. (other steps omitted) This is the current step: Wash Broccoli in water and
shake off excess water. Chop top (head) of the Broccoli into bite-sized pieces.
<|endoftext|> <|assistant|> ok! <|endoftext|> <|prompter|> Is it necessary to wash
the broccoli before chopping? <|endoftext|> <|assistant|> Yes, it is important
to wash the broccoli before chopping to remove any dirt or debris. <|endoftext|>
<|prompter|> next. <|endoftext|> <|assistant|> Heat Extra-Virgin Olive Oil in a
large nonstick skillet over medium-high heat. <|endoftext|> </s>
```

Table 10: Example of the input prompt to the Vicuna model.

state transition probabilities and enable the creation of a user utterance bank linked to specific state and event pairs.

The user simulation generation process starts with the tool initiating an interaction with TWIZ. For each turn of the conversation, an event is selected based on the computed transition probabilities and the current state of the dialogue. Then, the user utterances related to the current state and sampled event are ranked by similarity using sentence embeddings [28]. This ranking is performed by calculating the contextual embeddings of the ongoing conversation and determining the cosine similarity with the contextual embeddings of all potential utterances. This makes it so that the most relevant and contextually appropriate utterance is chosen, resulting in compelling and cohesive dialogues. An example is provided in Figure 8.

D.2 Rating Prediction for Conversational Task Assistants

Due to the complex interactions between the user and the system, errors are prone to happen which in turn lead to user dissatisfaction and low ratings. Being able to predict the rating of an interaction is thus a critical step to understand the system's shortcomings, and act accordingly [18, 5]. Emphasizing this, on average less than 5% of conversations have an associated rating, making it difficult to decide which conversations should be prioritized for analysis. Moreover, rating prediction complements the creation of the simulated interactions (Section D.1), allowing to automatically rate these generated interactions.

Inspired by the work from *Choi et al.* [5] in SocialBot, we developed a rating prediction model specific to the TaskBot setting [9], where the aim is to predict a rating given an entire conversation.

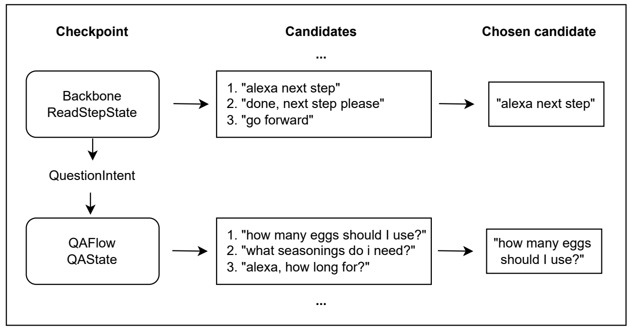

Figure 8: User simulation procedure.

Our model is based on the Transformer [38] architecture and combines both textual features, i.e., the conversation between the user and system, and user-behavior features, such as the number of fallbacks/steps read, and other general and Taskbot-specific features. To evaluate our model, we used conversations and ratings collected during the first edition of the Alexa Prize TaskBot challenge. In this setting, the model achieves an accuracy of 70% in a binary rating classification task. The results demonstrated the utility of automatic rating prediction, revealing insights such as the significance of starting a task and the increased importance of the latter part of the conversation [9].