

---

# GRILLBot-v2: Generative Models for Multi-Modal Task-Oriented Assistance

---

**Sophie Fischer**

University of Glasgow  
s.fischer.1@research.gla.ac.uk

**Niklas Tecklenburg**

University of Glasgow  
2465258t@student.gla.ac.uk

**Philip Zobel**

University of Glasgow  
2479229z@student.gla.ac.uk

**Eva Kupcova**

University of Glasgow  
2479253k@student.gla.ac.uk

**Ekaterina Terzieva**

University of Glasgow  
2403606T@student.gla.ac.uk

**Daniel Armstrong**

University of Glasgow  
2832539A@student.gla.ac.uk

**Carlos Gemmell**

University of Glasgow  
c.gemmell.1@research.gla.ac.uk

**Iain Mackie**

University of Glasgow  
i.mackie.1@research.gla.ac.uk

**Federico Rossetto**

University of Glasgow  
f.rossetto.1@research.gla.ac.uk

**Jeff Dalton**

University of Glasgow  
jeff.dalton@glasgow.ac.uk

## Abstract

We present our Alexa TaskBot Challenge virtual assistant, GRILLBot-v2, which advances multi-modal task-oriented conversations by leveraging generative language models and automatic extraction and augmentation of interactive task data. GRILLBot-v2 is a conversational system based on open-source software and publicly available data. The task of manually crafting engaging conversational content is expensive and time-consuming. Conversely, using solely generative models has the danger of hallucinations or forgetting conversational history. Therefore, we propose advances in task-oriented automated data extraction and augmentation to create multi-modal corpora, including tasks, domain knowledge, and associated videos. Specifically, we include rich content such as custom jokes, system-initiative questions, and multi-modal content. Furthermore, this rich data allows us to ground our generative models to create complex and interactive conversations. For example, we make meaningful improvements to task QA by grounding answer generation on conversation history and our knowledge and task corpora, achieving a factoid accuracy of 0.94 on our test set. We also use structured information within tasks (e.g., category tags for recipes) to create a mixed-initiative approach to guide users to more relevant search results, increasing the average conversation rating by over  $\sim 15\%$  (guided search) during the semi-finals. The success of GRILLBot-v2 during the competition motivates using synthetic content from generative models for tasks by grounding them on relevant knowledge and task data.

# 1 Introduction

Task-oriented assistants empower users to perform complex activities like cooking advanced recipes. They require a deep understanding of world knowledge to answer possible user questions. Moreover, these assistants must present very dense information without overloading users. To this end, next-generation virtual assistants will leverage visual and auditory cues to interactively and fluently guide users through highly complex tasks. They require incorporating images and videos to teach and demonstrate critical techniques or actions necessary to perform the user activity. Generative Large Language Models (LLMs) make this possible by following highly specific and curated user instructions [Touvron et al., 2023b].

Despite recent advances in LLMs, there are multiple open challenges to developing a task-oriented multi-modal conversational agent. For example, finding relevant tasks for each user is complex due to the user’s environment and preferences. Furthermore, task-oriented agents based on generative models are rarely multi-modal, with few exceptions like GPT4 [OpenAI, 2023]. This is despite previous research showing that aligning multi-modal content to assist the user in performing complicated tasks can increase the learning and success rate [Fischer et al., 2022]. In addition, generative models are expensive to run at scale and can be unpredictable, potentially generating incorrect or inappropriate responses [Gehman et al., 2020].

To address these challenges, we propose GRILLBot-v2, a task-oriented multi-modal conversational agent developed during the 2022/2023 Alexa Prize TaskBot Challenge [Agichtein et al., 2023]. The TaskBot Challenge aims to empower users to complete real-world complex tasks, like preparing a BBQ from a specific region in the US or painting a wall based on specific environmental information like climate, available materials and skill set. These tasks are complex due to the user-specific and technically challenging skills required for completion. To support complex tasks in a fluent, fact-checked, and knowledgeable manner, we propose to advance the field in the following ways.

First, we develop methods to extract, augment, and synthesise task data to generate engaging conversational content. In particular, we enrich our TaskGraph Corpus [Gemmell et al., 2022], which models tasks into a dynamic graph-based representation, with offline augmentations using generative models. Furthermore, we use our task-based Knowledge and Action Video Corpora to help dynamically ground our conversations to external knowledge and multi-modal content. Our offline augmentations include adding information from related blog posts, descriptions, jokes, and fun facts. Additionally, we develop a video alignment process that transcribes, segments, and aligns video snippets to tasks using sequence-to-sequence models. We also research running these augmentations as part of the live system using generative models [Dettmers et al., 2023, Taori et al., 2023, Hu et al., 2021].

Second, GRILLBot-v2 handles multi-modal interactions and complex conversational flows around users performing tasks. We employ multiple component-level LLMs that we ground to conversation history and task corpora. A primary challenge is understanding user intent and orchestrating system behaviour to flexibly solve the user’s information needs. Extending previous work [Gemmell et al., 2022], we advance generative state transitions based on the conversation state and history with a sequence-to-sequence model called Neural Decision Parser. We address previous issues with hallucinations by using a new model and training data, achieving an intent accuracy of 0.83 on our test set.

Third, we automatically generate guided search flows to help the user decide which task to choose. We base guided search flows on grouped topically similar tasks called “categories”. By creating thematic trajectories, even users with vague queries can effectively search for tasks that align with their specific goals and preferences. Our average user rating increased from 3.03 to 3.28 during the semi-finals when vague users selected the guided search flow.

To create a fluent assistant that understands, acknowledges, and adapts tasks based on user input, we deploy a relatively lightweight generative model live [Taori et al., 2023]. We ground generation in conversational context to avoid hallucinations and develop methods to check generated content for inappropriateness. This allows advanced chit-chat and question-answering, which are essential to a genuinely engaging, knowledgeable, and useful task-oriented conversation. We provide strong coverage of user questions, including task-specific QA, knowledge-augmented QA, and LLM QA, which is purely generative. Furthermore, we include generative approaches incorporating more socially engaging elements, including jokes and system-initiative questions. These advances allow

us to have long and interactive conversations. During the semi-finals, users who utilise the system version featuring generative model engage in conversations that are 37% longer (around 20.1 turns with the generative model compared to 14.6 turns without it).

We use data and insights from several months of deployment on Alexa’s US voice-only and screen-based devices to further evaluate our approaches. We perform daily log analysis to observe user behaviour and build processes to train system components based on annotated user logs. We furthermore will be releasing a new edition of GRILLBot-v2 as part of our open-source software OAT framework for usage in the community <sup>1</sup>.

To summarise, this work makes several novel contributions:

- Methods to synthetically generate and enhance complex tasks offline for engaging conversational content.
- An improved model for generating accurate system actions flexibly for unseen user requests.
- Methods to generate synthetic thematic trajectories for user recommendation and search guidance.
- Complex QA and chit-chat capabilities using knowledge sources and LLMs.

## 2 Conversation Design

GRILLBot-v2 enables learning and task assistance for real-world activities at home and in the kitchen. The goal is to assist the user with personalised, interactive and adaptable tasks to be truly helpful on a daily basis.

We define four conversation stages:

- **Welcome:** We greet the user, introduce GRILLBot-v2’s capabilities, and recommend currently promoted curated tasks.
- **Search:** Based on user input, we search for a suitable task or offer recommendations depending on the user’s information need.
- **Selection:** GRILLBot-v2 explains why it shows search results and compares them to help decision-making. The user can preview task summaries highlighting specific requirements like ingredients and skills required. The user can also modify and refine the task as needed.
- **Execution:** The user has selected a task. GRILLBot-v2 walks the user through the instructions interactively. We offer additional tips, facts, jokes, frequently asked questions, helpful videos in this phase, and check-in with the user during the task progression.

<sup>1</sup><https://github.com/grill-lab/OAT>



Figure 1: Extract of conversations with multi-modal jokes relevant to current task in execution.

To help users decide which task is most relevant, we implement support to guide users to specific results by incorporating automatically generated task category content using LLMs. Section 6 describes the LLM integration in the system, which helps GRILLBot-v2 to recognise, acknowledge, and respond to user queries.

Furthermore, to create engaging and helpful tasks, we transform standard linear tasks to be more entertaining and engaging. Given the challenges of conveying detailed and technically complex task steps through voice alone [Frummet et al., 2019, Desai and Chin, 2021], our solution relies heavily on multi-modal screens whenever they are accessible. Prior research has demonstrated that strategically incorporating helpful videos can empower individuals without the expertise to acquire new skills and effectively accomplish challenging tasks [Fischer et al., 2022]. Therefore, we create task augmentations offline to link external content to tasks. Figure 1 shows examples of multi-modal content in the form of jokes that GRILLBot-v2 shares with the user during the conversation. In addition, we enhance tasks by including additional information, fun facts, and multi-modal elements, such as videos and images.

Users ask any questions during the task execution and beyond. To respond fluently, GRILLBot-v2 uses LLMs and knowledge corpora to ground the answer in the user’s past conversations and real-world knowledge. The system fluently declines out-of-scope requests, generates chit-chat answers, and asks clarifying questions.

### 3 Data Pipelines and Augmentations

The primary objective of GRILLBot-v2’s data pipeline is to shift the pre-processing of knowledge sources to an offline stage to reduce system latency during live conversations.

Figure 2 shows the four main data contributions that the offline pipeline produces. The first is the augmented TaskGraph corpus, which GRILLBot-v2 uses for task search and recommendation. Second, to support guided search, we create a Category Corpus consisting of topically grouped tasks which GRILLBot-v2 uses to find relevant results for users searching for vague queries. Third, we collect a Knowledge Corpus from pages linked by task articles and create extra information knowledge documents for task QA. Lastly, we collect multi-modal data for our Video Action Corpus. We describe these in more detail in the following sections.

#### 3.1 Augmented TaskGraph Corpus

Based on previous work, we use TaskGraphs to model task structure and enable complex flows of live conversations [Gemmell et al., 2022]. TaskGraphs are directed acyclic graphs that model actions and information dependencies that the system uses to aid the user in performing the task. Information is

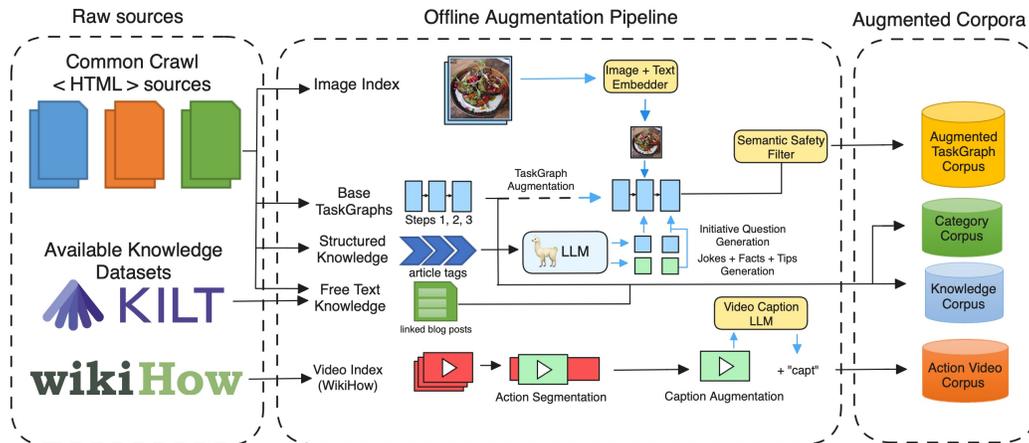


Figure 2: Offline pipeline: Converting raw task sources to create engaging TaskGraphs.

represented with heterogeneous nodes, each with a specific role, i.e. steps, requirements, conditions, logic, action, and extra information.

Figure 2 shows how our pipeline converts raw textual task sources (Common Crawl, Wikipedia, WikiHow) into an augmented TaskGraph corpus using LLMs, external knowledge sources, and multi-modal content. Specifically for Common Crawl, we retrieve the HTML for specific domains and build wrappers to extract the task information. Table 1 shows the domains that our TaskGraph corpus contains.

Table 1: Distribution of augmented TaskGraphs across different domains.

domain	task count
epicurious.com	496
food52.com	10,165
foodnetwork.com	8,499
seriouseats.com	5,317
wholefoodsmarket.com	2,675
wikihow.com	63,004
total count.com	90,156

Table 2: Comparison of Category, Subcategory, and Task Counts Between the Cooking and DIY Domains.

domain	Cooking	DIY	Total
categories	547	344	891
subcategories	1,945	1,641	3,586
linked TaskGraphs	5,475	12,689	18,164

We then pass our extracted raw task information into a suite of “augmenters” that transform text-based tasks into multi-modal engaging experiences. Specifically, task augmentations aim to translate, adapt and decompose the task into meaningful executable units. Furthermore, the augmented TaskGraphs allow complex conversational paths depending on user preferences, abilities, and requirements.

Figure 3 show an example of an augmented TaskGraph. Our augmentations access the knowledge corpus to include images, aligning video snippets, multi-modal jokes, and relevant extra information. Augmentations that use generative models [Taori et al., 2023] can reformulate task steps, generate task descriptions, and produce system initiative questions to check in with the user during the task as well as answers to common questions. For multi-modal jokes, we hand-curate a set of jokes to match task step keywords (such as common ingredients) and synthesise matching images using diffusion models [Rombach et al., 2022]. Augmentations improve the tasks by effectively communicating clear and meaningful executable information units to the user.

Prior work shows that it is important to complement voice instructions with visual information to assist the user and support their learning experience [Fischer et al., 2022]. For recipes that do not have

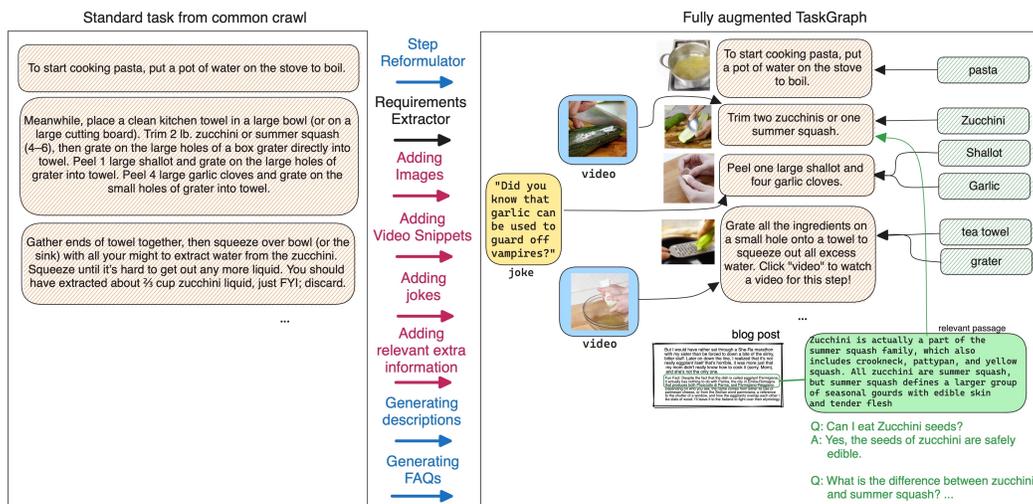


Figure 3: Converting linear text-based tasks into multi-modal non-linear TaskGraphs allows accessing and scheduling linked components live during the conversation. Blue augmentations involve generative models, and red augmentations access the knowledge corpus.

step images, we perform image retrieval [Radford et al., 2021] to augment task steps by associating them with relevant images sourced from our comprehensive image corpus, containing all images found in the task corpus.

To enhance the multi-modal experience, we extract 10,885 videos [Zhang et al., 2020] during the offline pipeline and use action segmentation, vision properties, and audio to align videos to task steps with an offline video alignment augments using various off-the-shelf models [Radford et al., 2023, Reimers and Gurevych, 2019]. Overall, this results in 90,156 augmented TaskGraphs that the live system uses during task search.

### 3.2 Category Corpus

We automatically generate guided search flows based on grouped topically similar tasks to help the user issuing vague queries decide. A group of topically similar tasks is called a category. Individual subcategories and task options are crawled from breadcrumbs on target domain webpages (serious-eats.com for cooking and wikihow.com for home improvement) and consolidated into categories.

For cooking, this results in categories for specific courses (e.g. breakfast), ingredients (e.g. beans), cuisine (e.g. Mexican), diets (e.g. gluten-free), and holidays & seasons (e.g. Independence Day). We then crawl the most mentioned tags on each category’s overview page. As a result, we achieve a hierarchical structure starting with the main topic, followed by subcategories derived from the tags, and finally, directly linked recipes. On the other hand, WikiHow uses breadcrumbs to link to categories matching articles, suggesting related topics that contain tasks. For home improvement, categories are terms such as "arts & crafts" and "gardening" that can describe a variety of tasks.

Our Category corpus contains 891 categories (547 cooking domain, 344 wikiHow). Each category has multiple subcategories with multiple linked tasks each, resulting in 18,164 unique tasks as part of the Category Corpora. Table 2 shows the total number of subcategories and linked TaskGraphs for each domain. On average, we have four subcategories for each category. For each subcategory there are 5 task candidates.

### 3.3 Knowledge Corpus

During the TaskGraph Corpora creation, TaskGraph wrappers discard content such as related blog posts, metadata, and multi-modal content that are difficult to parse into or align to a TaskGraph. Especially Serious Eats articles contain a lot of high-quality content that we would otherwise discard. Therefore, we use Serious Eats for building the corpus as the main source for high-quality content in the cooking and recipes domains.

The knowledge corpus comprises 18,241 knowledge documents extracted from 10,750 Serious Eats articles. Articles contain additional explanations for recipes, including Wikipedia-like information with background science, history, and facts. The articles relate to recipes in the corpus and fundamental cooking techniques.

We index the relevant metadata of each paragraph, including the article’s author, title, and publication date. The applications of this knowledge corpus are versatile. In both live and offline systems, it enhances question-answering processes and augments individual TaskGraphs. Beyond improving QA for cooking tasks, we leverage links between the Serious Eats TaskGraphs and articles to reinforce the relevance of associated context, expanding its usability across various domains.

### 3.4 Action Video Corpus

We extract videos for multi-modal task augmentation [Zhang et al., 2020] during the offline pipeline. An augments then aligns videos to task steps using action segmentation, vision properties, and audio transcription. The offline corpus currently consists of 6,112 cooking videos, 2,132 how-to videos, and 4,473 Wikihow videos, making a total of 10,885 videos. The video step augments module uses this corpus to automatically associate relevant video snippets with corresponding steps in the task.

In previous work, we collected a video corpus containing 733 general Wiki- How “How-To” videos since most task sources do not include fine-grained rich multi-modal content for each task step [Gemmell et al., 2022]. These videos can generalise and match to different task steps that do not contain linked videos [Fischer et al., 2022].

## 4 Online System

In this section, we explain the online components of GRILLBot-v2, which enable walking users through complex tasks. Based on previous research [Gemmell et al., 2022], the *Orchestrator* components direct the system behaviour using different micro-services. Figure 4 shows our online micro-services architecture.

### 4.1 System overview

When the user first interacts with GRILLBot-v2, we classify the topic domain of their request [Gemmell et al., 2022]. Next, a new *Neural Decision Parser* classifies the user utterance to generate an appropriate system action based on the task and conversation context. The live system guides the user through a task by scheduling different nodes of the TaskGraph depending on the intent classification and arguments generated from the Neural Decision Parser. Subsequently, the internal system scheduling processes this request and efficiently schedules content for system response, which GRILLBot-v2 presents to the user through populating the screen and voice output.

GRILLBot-v2 shows relevant content when possible but also acknowledges what the user has said and keeps them within the boundaries of system capabilities. The offline pipeline ensures engaging and presentable content by creating augmented TaskGraphs. However, even if we run relatively small generative models as part of the offline pipeline, they are still expensive to run on the whole corpus. Therefore, we also build an online task augmentations micro-service called *Task Enhancer*. This service starts running when the user is in the Selection phase, allowing enhancing the task options in the background. This service allows spending resources only on tasks that users select, and since we cache enhancements, we iteratively improve our TaskGraph corpus. We also support rewriting task steps, synthesising descriptions, and generating system initiative questions based on the task step.

### 4.2 Deployment details

GRILLBot-v2 runs in distinct containers on a dockerized Kubernetes EKS (Elastic Kubernetes Service) cluster hosted on AWS. Our system architecture enables the dynamic scaling of various micro-services based on demand Kubernetes load-balancing features. Additionally, this setup allows us to deploy a generative model as a micro-service in the live cluster without slowing other micro-services down. Some other system components, such as neural IR techniques [Lin et al., 2021], also require GPU access, which we can assign appropriately.

In the following sections, we describe the remaining micro-services, including the Neural Decision Parser (NDP), search and recommendation, and knowledge question answering & chit-chat.

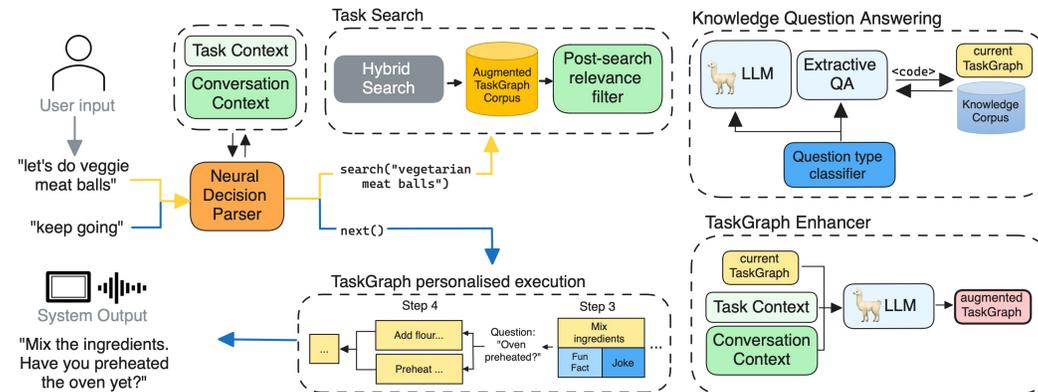


Figure 4: Online system micro-services overview. Different micro-services are called during user interactions to generate an appropriate system response.

Table 3: Evaluation Metrics for Neural Decision Parser V1 and V2

Version	Metrics			
	Precision	Recall	F1 Score	Accuracy
NDP V1	0.767	0.803	0.768	0.798
NDP V2	0.841	0.788	0.801	0.830

## 5 Generative System Actions

GRILLBot-v2 uses a generative model known as the Neural Decision Parser (NDP) [Gemmell et al., 2022] to comprehend user commands in natural language. The NDP acts as a system action generator, converting input into system actions using a Domain Specific Language (DSL). This DSL consists of predefined system commands, allowing for flexibility with parameterisation. Unlike traditional intent classifiers, the NDP uses the user’s conversational history for contextualised semantic parsing.

Our new Neural Decision Parser is based on a FLAN-T5 model [Chung et al., 2022]. We train the model on a new dataset that leverages manually annotated user logs. To prevent skewing the model towards specific intents, we generate synthetic conversation data to balance the intents distribution using ChatGPT [OpenAI, 2023]. The resulting dataset contains 1,240 samples (split into 80% training, 10% validation, and 10% testing). Each sample in the dataset contains the current user utterance, the previous system response, and potential system actions, providing context-rich information for effective handling of coreference resolution and search parameterisation.

Table 3 reports the performance metrics between NDP version 1, used during last year’s Alexa TaskBot Challenge [Gemmell et al., 2022], and the current NDP (here referred to as version 2). NDP V2 outperforms NDP V1 by achieving an F1 score of 0.801 on the test set. Furthermore, NDP V2 exhibits improved accuracy with a rate of 0.830, surpassing NDP V1’s accuracy rate with an absolute 3% improvement.

In the appendix, we also report the model metrics on the test set broken down by each possible user action in Table 5. Notably, “chit\_chat()” attains the lowest F1 score (0.453) due to its diverse, open-ended nature, absence of clear patterns, and informal style.

## 6 Search & Recommendation

To find relevant task graphs, we address the search process in three ways. Each focuses on a different kind of user need. GRILLBot-v2 implements searching for users with specific ideas (“spaghetti bolognese”) and vague ideas (“I want to have breakfast”) and for users who don’t know what to search for. The NDP-generated system action includes the different search requests to allow issuing the search query to the search modules.

We define the types of information finding as follows.

- **Specific search:** If the user asks for a specific task (a specific information need), task search should retrieve a ranked list of TaskGraphs ( $[T_1, T_2, \dots, T_N]$ ) from a corpus ( $CT$ ) given an information need ( $Q$ ).
- **Category search:** If the user asks for something vague that matches a category, we insert an elicitation step. Task search should retrieve a matching candidate  $C$  with subcategories  $S$  where each subcategory contains a list of TaskGraphs ( $[T_1, T_2, \dots, T_N]$ ) from a category corpus ( $CT$ ) given an information need ( $Q$ ).
- **Recommendation:** If the user provides no information need, we recommend a list of seasonally hand-crafted TaskGraphs ( $[T_1, T_2, \dots, T_N]$ ).

Figure 5 shows example conversation flows for the three different ways of searching.

The initial retrieval of specific and vague searches uses the reciprocal rank fusion technique [Cormack et al., 2009] of two models (Colbert v2 [Lin et al., 2021] and BM25 [Croft et al., 2010]), in conjunction with query expansion using RM3 [Abdul-Jaleel et al., 2004] on the TaskGraph corpus. We perform

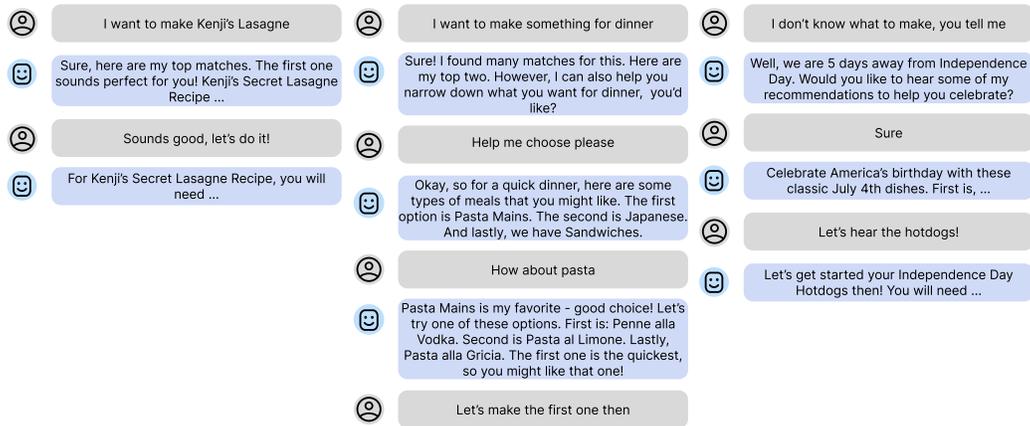


Figure 5: Three different conversations modeling how users get to task execution: a user who knows what they want help with, a vague user walks through a category-specific, automatically curated elicitation process and third season-specific recommendations for unsure users.

hyperparameter tuning for BM25 and RM3 to optimise the retrieval performance. The evaluation uses the `recipe1mln+` [Marin et al., 2019] and `wikiHow` [Zhang, 2022] datasets. We employ k-fold cross-validation to optimise the following parameters: length normalisation factor ( $b$ ), term saturation parameter ( $k$ ), number of feedback documents (`fb_docs`), number of feedback terms (`fb_terms`), and the weight assigned to the original query in RM3. Furthermore, we query Amazon’s third-party APIs for each query, resulting in a diverse pool of candidates.

To address new users unfamiliar with the system and lacking specific ideas for tasks, we provide seasonal recommendations. We automatically generate tasks for all American holidays using ChatGPT [OpenAI, 2023]. For example, GRILLBot-v2 suggests the “Independence Day” category to support the user in planning their perfect get-together. We share our prompts in Appendix A. By asking the LLM to generate Python dictionaries, we can easily import the generated data into our custom annotation tool to decide whether the synthetic task is valid. The annotation process still needs a human in the loop. However, the data curation process is much faster since we only have to review content, not curate. This feature allows us to present highly curated, up-to-date content to our users without high manual human effort.

## 7 Knowledge Question Answering & Chit-Chat

A knowledgeable and helpful multi-modal assistant needs domain knowledge to provide relevant and fluent answers. Even if the answer to a request is not known or not supported, GRILLBot-v2 needs to acknowledge and react to user requests beyond just walking through the task. Therefore, our QA and chit-chat components play a crucial role in reliably engaging the user with GRILLBot-v2.

We approach this problem with a composed QA and chit-chat system. The composed system uses the user’s past interactions to ground the conversation in real-world knowledge using LLMs and knowledge corpora. Based on previous research [Gemmell et al., 2022], user requests are classified into different QA types using the NDP. We group question types into factoid questions, where the model extracts the answer from the context provided, and open-ended questions that require the model to leverage external knowledge sources.

Our different models in the composed QA & chit-chat system are as follows:

**Flan T5-based QA** GRILLBot-v2 uses this QA module for task-specific questions where the task information contains the answer. It is based on a FLAN-T5 [Chung et al., 2022] model and relies on the conversation context to extract the most relevant paragraph. To provide further relevant content to the model, we leverage the knowledge corpus for retrieval-augmented answers in the cooking domain. Given an instruction in the prompt and the context, we zero-shot prompt the model to extract the information to provide a relevant response to the user.

```

### Instruction: You are a friendly AI assistant who is assisting a human making {task_title}.
Based on the current recipe step, give a very short answer to the user question. This is the
current step: {context}.
### Input: User: {user_question}
### Response: Your response:"

```

Figure 6: A sample prompt to the Alpaca 7B model for step specific question answering. The task title, step context and user question are substituted.

Table 4: QA model performance on our 200 questions test set, containing “easy” factoid questions and “hard” generative questions. Each model answer’s relevance is evaluated by human annotators.

		Guanaco	Alpaca 7B	Alpaca Lora	FLAN T5
accuracy	factoid	0.860	0.940	0.900	0.890
	generative	0.938	0.927	0.923	0.365
	combined	0.898	0.934	0.913	0.633
response time (in sec)		4.212	0.866	3.122	0.412

**Alpaca 7B LLM-based QA** This module is used for both QA and chit-chat and is based on the LLaMA 7B model [Touvron et al., 2023a]. Taori et al. [2023] trained the base model on various instruction tuning datasets including 52k instruction-following demonstrations; therefore, it generalises well for both QA and chit-chat response generation. We using hand-crafted contextualised prompts depending on the QA classification type and zero-shot prompt the model to generate a relevant response to the user question. Figure 6 shows an example prompt to the Alpaca LLM. The appendix contains additional prompts, example questions and answers generated by the different models for comparison.

To evaluate how the different models perform on typical user questions, we create a 200-sample test set split into open-ended and factoid questions. We use reformulated logs and synthetically generated questions and answers using ChatGPT [OpenAI, 2023] to collect question-answer pairs for the dataset. Given the synthetically generated tasks for the theme content (Section 6), we prompt the model to generate factoid and open-ended questions about the task step. The dataset is also split into factoid and generative questions. Figure 8 shows example questions.

Table 4 shows the performance of each model’s responses based on human relevance annotations. The annotator is given the ChatGPT answer as guidance for a relevant response. We compare different instruction-following open-source LLMs, including Guanaco [Dettmers et al., 2023], Alpaca 7B [Taori et al., 2023] and Alpaca Lora [Wang et al., 2023], which are all fine-tuned from Meta’s LLaMA model [Touvron et al., 2023a]. We observe very similar model performance on the factoid questions, where models are able to extract the answer from the given step text or conversation history. However, for generative questions, we see that models like FLAN T5 [Chung et al., 2022] are outperformed by the LLaMa models by ~ 56%.

Based on the evaluation, we use FLAN T5 and Alpaca 7B in the live system. These models are the best-performing models with the shortest latency. Figure 7 shows our composed QA architecture. FLAN T5 answers factoid question types such as task-specific, step-specific, and task-comparison questions that the model can answer based on the context. Since all other questions require more knowledge, user acknowledgement, personality, and opinions, Alpaca 7B LLM performs well due to its quick response time and ability to follow prompt instructions. We can achieve highly fluent contextualised responses in less than 1.5 seconds by limiting the number of tokens generated. For safeguarding, we check the generated responses for sensibility and appropriateness. Additionally, we implemented a timeout to ensure timely responses.

For chit-chat, if the utterance does not match several personality chit-chat FAQs, we query the Alpaca 7B LLM. The LLM real-world knowledge and fluency enable GRILLBot-v2 to respond to a wide range of topics the user wishes to discuss, ensuring a more engaging conversation experience. However, we also decline out-of-domain requests by acknowledging user requests and guiding users

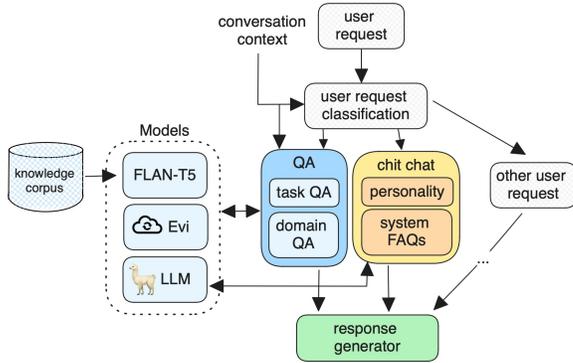


Figure 7: Composed QA and chat system. Depending on question type, micro-services are called.

**Factoid questions:**  
 What temperature should the oven be for preheating?  
 How should I cut the garlic?

---

**Generative questions:**  
 How many tomatoes can I expect per plant?  
 How do you know that the slider patties are cooked?

Figure 8: Examples of factoid and generative questions from the QA eval dataset

to the core capabilities of GRILLBot-v2. For any utterances classified as “Confused User”, we acknowledge the request and offer rerouting to the help corner.

## 8 Analysis

To evaluate the success of individual conversations, we focus on key performance indicators, including user ratings and conversation length. Furthermore, we perform a manual log analysis to gain further insights. Using a custom dashboard, we perform daily term and conversation-level annotations. By rating samples of user-rated conversations daily, we gain insights into our perceived quality of the currently live system. Figure 9 compare the average daily rating of our versus the user’s rating. We observe that user ratings are noisy. However, we observe a positive trend in our and the users’ ratings. In Appendix D, we include further analysis of annotated conversations.

For individual system components, we annotate selected specific conversations for training data. For example, this enables us to retrain the NDP to achieve higher accuracy. We rephrase user questions to include in our QA test set. To gain insight into performance on a turn level, we are working on integrating a turn-level sentiment classifier into the dashboard. The model calculates a turn-level CSAT to predict the likeliness of unsatisfied users and will help us analyse large volumes of user data.

Using the log analysis, we estimate the impact of new features in the system. According to our annotated CSAT, synthetic content generation offline improves the TaskGraph quality and, therefore, the conversational experience. Furthermore, 7-day Moving Average (L7d CSAT) user ratings also increase from 2.88 to 3.19 in semi-finals since GRILLBot-v2 recommends AI-generated themes. Conversations in which users who opted for the AI-generated themes achieve an even higher average rating of 3.42.

Across the semi-finals, many users are first-time users. Using the chit-chat and QA strategies described in Section 7, we can inform the user of GRILLBot-v2’s capabilities and answer requests beyond navigation effectively. Specifically, conversation length increased significantly since deploying generative models in the live system. The conversation length, including chit-chat or QA, increases from an average of 14.5 turns to 20, showcasing that GRILLBot-v2 can recover confused users and engage them in conversation. Reviewing user conversations, we note that introducing the Alpaca LLM impacts the system’s fluency, which motivates future research into fully synthetic generative task-oriented conversational agents.

During the semi-finals, we experiment by introducing a synthetically generated guided search flow using categories for users that utter vague queries. The guided search flow helps users decide what tasks to do and meet their expectations. Throughout the semi-finals, about 11% of the users issue vague searches and enter the category search flow. We achieve an average user rating of 3.75 vs. 3.26 if a vague user enter the search flow compared to if they decide to select a possible unrelated result. The positive feedback motivates extending preference elicitation across specific search queries.

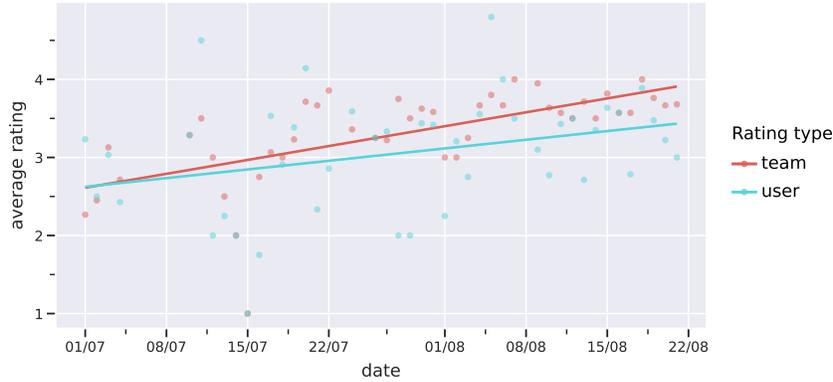


Figure 9: GRILLBot-v2’s Average daily ratings on an annotated subset of conversations. Blue shows user ratings, red shows what the team rated conversations with.

Lastly, we enhance the user experience using multi-modal augmentations during task execution. The offline pipeline adds images, fun facts, and jokes and aligns videos. Given our fast-paced and dynamic deployment approach throughout the competition, it becomes challenging to measure the precise impact of these features independently from other factors. However, the difference in user ratings between devices with screens and headless devices during semi-finals (3.27 vs. 3.05 L7d CSAT score) suggests that multi-modal augmentations do contribute to enriching the overall user experience, making the system more enjoyable and relevant for users.

## 9 Conclusion

We present the second edition of GRILLBot-v2, a multi-modal agent grounded in real-world knowledge via generative models and additional knowledge sources. Developed based on vast user interactions with complex real-world tasks, GRILLBot-v2 is a scalable conversational system using open-source software and non-proprietary data. Our advancements lie in task-oriented automated data extraction and augmentation using generative models. We enhance these conversations with interactive elements like custom jokes and multi-modal content, grounding conversations in our tasks, knowledge, and video corpora. We also enhance search quality for vague queries by using structured information within tasks, such as category tags for recipes. Furthermore, we show that synthetic content increases average user ratings without needing the manual creation of data. Additionally, we improve task QA by leveraging different LLMs, ensuring answer generation aligns with conversation history and our knowledge and task corpora to increase conversation length significantly.

As part of the journey towards multi-modal agents that guide and engage users through achieving real-world tasks, our findings motivate further research into task-oriented synthetic data generation. Specifically, generated categories to create a guided search flow and generated tasks lead to a better user experience. The alignment of multi-modal contexts for complex tasks has room for improvement. Furthermore, we want to scale up generating fully synthetic tasks beyond offline creation. Specifically, the goal is to create and adapt tasks online to suit user needs including image input for meaningful interaction beyond voice.

## Acknowledgements

The authors would like to thank the following researchers for their support and collaboration: Sarah Rose Siskind, Tom Koff and Mary-Ellen Foster. This work is supported by a Turing AI Acceleration Fellowship from the Engineering and Physical Sciences Research Council, grant number EP/V025708/1.

## References

- N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. Umass at trec 2004: Novelty and hard. *Computer Science Department Faculty Publication Series*, page 189, 2004.
- 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. Rucker, 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>.
- H. W. Chung, L. Hou, S. Longpre, B. Zoph, and Y. Tay. Scaling instruction-finetuned language models, 2022.
- G. V. Cormack, C. L. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759, 2009.
- W. B. Croft, D. Metzler, and T. Strohman. *Retrieval Models*, page 250–251. Pearson Education, 2010.
- S. Desai and J. Chin. Hey google, can you help me learn? In *Proceedings of the 3rd Conference on Conversational User Interfaces*, pages 1–4, 2021.
- T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *ArXiv*, 2023.
- S. Fischer, C. Gemmell, I. Mackie, and J. Dalton. Vilt: Video instructions linking for complex tasks. In *Proceedings of the 2nd International Workshop on Interactive Multimedia Retrieval*, pages 41–47, 2022.
- A. Frummet, D. Elswiler, and B. Ludwig. Detecting domain-specific information needs in conversational search dialogues. 2019.
- S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. 2020.
- C. Gemmell, S. Fischer, I. Mackie, P. Owoicho, F. Rossetto, and J. Dalton. Grillbot: A flexible conversational agent for solving complex real-world tasks. *1st Proceedings of the Alexa Prize Taskbot*, 2022.
- 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. *ArXiv*, 2021.
- S.-C. Lin, J.-H. Yang, and J. Lin. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (ReplANLP-2021)*, 2021.
- J. Marin, A. Biswas, F. Offi, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.
- OpenAI. Gpt-4 technical report. 2023.
- 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. In *International Conference on Machine Learning*, 2021.
- A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, 2023.
- N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *ArXiv*, 2019.
- R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.
- R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto. Stanford alpaca: An instruction-following llama model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca), 2023.
- H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *ArXiv*, 2023a.

- H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, 2023b.
- E. J. Wang, C. Alexiuk, Y. Bo, Z. Yang, K. Kwok, I. Gusev, A. Echavez, et al. Alpaca-lora: Instruct-tune llama on consumer hardware. <https://github.com/tloen/alpaca-lora>, 2023.
- L. Zhang, Q. LYU, and C. Callison-Burch. Intent detection with wikiphow. In *AAACL*, 2020.
- L. H. Zhang. wikiphow-goal-step. <https://github.com/zharry29/wikiphow-goal-step>, 2022.

# Appendices

## A Synthetic Content Generation with ChatGPT

Imagine you are an Alexa assistant who is helping the user with cooking and home improvement tasks. The user has been searching for {task\_name} and you need to come up with a guide that is engaging, entertaining and fun. Bear in mind the user actually wants to do the task.

Fill in the following fields with your task details:

- title
- thumbnail\_image (detailed image description for image creation with stable diffusion)
- description (2 sentences max.)
- materials (list of dictionaries): Each material contains:
  - name
  - amount
- steps (list of dictionaries) [up to 10 steps]: Each step contains:
  - speech\_text that is read to the user (no more than 3 sentences)
  - screen\_text that is shown on the screen and gives more details about the step
  - step\_image (description of a simple and minimalistic step image)
  - alt\_image (description of an image containing one item mentioned in the step)

Return your answer as a python dictionary. Only respond with the dictionary.

Figure 10: ChatGPT Synthetic Task Generation Prompt to generate tasks offline, which we can convert and add to our TaskCorpora. We use stable diffusion models to synthesise the step images and thumbnail images. We review all generated tasks and images as part of the pipeline.

I need help with all upcoming American food holidays (like National Pizza Day) and public holidays in 2023. Popular tasks should include at least five recipes if applicable and five tasks that you could do at home, like crafting, home improvement, workouts and gardening. Generate examples in the below Python format:

```
{
  'holiday_name': {
    'popular_tasks': [],
    'popular_recipes': [],
    'description': "",
    'date': "%d-%m-%Y",
  },
  {...}
}
```

Figure 11: ChatGPT Synthetic Theme Generation Prompt. We use this to recommend relevant date-specific themes to the user on the splash screen. Furthermore, GRILLBot-v2 shows these recommendations if the user is unsure what to search for and asks for help. We also wish the user a Happy {theme} if there is a valid theme and mention it as fun facts for food holidays.

## B Alpaca Prompt & QA Examples

Step specific question (hard)

**Alpaca 7B Prompt:**  
### Instruction: You are a friendly AI assistant who is assisting a human making Profiteroles. Based on the current recipe step, give a very short answer to the user question. This is the current step: To serve, using a serrated knife, slice each choux puff in half to create a top and bottom bun. Place a scoop of ice cream on each bottom half. Cover with the top half of each puff. Serve profiteroles immediately with chocolate sauce drizzled on top.  
### Input: User: *What other toppings or sauces can be used on the profiteroles?*  
### Response: Your response:

---

**Alpaca 7B response:** Other toppings or sauces that can be used on profiteroles include fresh fruit, whipped cream, chocolate sauce, caramel sauce, and nut butter.  
**Guanaco:** There are many toppings and sauces that can be used on profiteroles. Here are some ideas: 1. Chocolate: You can top your profiteroles with melted chocolate, chocolate syrup, or chocolate shavings. 2. Nutella: A popular choice for many, Nutella is a chocolate ...  
**Alpaca Lora:** Other toppings or sauces that can be used on the profiteroles include whipped cream, caramel sauce, chocolate sauce, and fruit compote.  
**FLAN T5:** chocolate sauce

Figure 12: Example of a “hard” step specific question, where the answer can’t be extracted from the text, but has to be generated. The question is *curative*. On the top is the Alpaca 7B prompt. Below are the answers that each system gave. Other system specific prompts are omitted for space reasons.

Step specific question (hard)

**Alpaca 7B Prompt:**  
### Instruction: You are a friendly AI assistant who is assisting a human with How to Grow Tomatoes Upside Down. Based on the current recipe step, give a very short answer to the user question. This is the current step: Step 4. Add a few drops of water. The goal is to moisten the new soil surrounding the seeds. You can use an eyedropper to apply a small amount of water, or wet your fingers and let a few drops of water drip off. Don’t add too much water, as the soil was already pre-moistened before planting. Keep the soil moist, but not soggy, as the sprouts grow. When the top of the soil becomes dry, add more water.  
### Input: User: *I added too much water I think what can I do?*  
### Response: Your response:

---

**Alpaca 7B:** You can try to drain some of the excess water by tilting the container and pouring it.  
**Guanaco:** Step 6. Increase  
**Alpaca Lora:** You can add more tomato cuttings to the pot and reduce the amount of water.  
**FLAN T5:** Add a few drops of water

Figure 13: Example of a “hard” step specific question. We observe that all responses apart from the Alpaca response do not really make sense.

Task specific question (hard)

**Alpaca 7B Prompt:**  
 ### Instruction: You are a friendly AI assistant who is assisting a human making Pizza Dough. Based on the current recipe step, give a very short answer to the user question. This is the current step: Step 5. Roll or pat out a piece of dough into a 12 inch circle, about 1/8-inch thick. Dust a pizza paddle with flour and slide it under the pizza dough. Brush the crust with a thin layer of olive oil, and top with your favorite flavors. Slide the pizza onto the hot stone in the oven and bake for 12 to 15 minutes, until the crust is golden and crisp. Repeat with the remaining dough.  
 ### Input: User: *I do not have a pizza paddle what can I use instead?*  
 ### Response: Your response:

---

**Alpaca 7B:** You can use a rolling pin or a large spoon to roll out the dough.  
**Guanaco:** You can use a baking sheet or a large plate. Just make sure that it is oiled or sprayed with cooking spray to prevent the dough from sticking.  
**Alpaca Lora:** You can use a large baking sheet or a pizza stone.  
**FLAN T5:** Flour

Figure 14: Example of a “hard” task specific question.

Selection question (hard)

**Alpaca 7B Prompt:**  
 ### Instruction: You are a friendly AI assistant who is assisting a human. You are helping the user choose between the following options: Option 1 is Search for Delicious Pizza by Maggie Hoffman, which takes 3 minutes and has a rating of 4.8 out of 5. Option 2 is Pizza Dough by Tyler Florence, which has no time estimate and has a rating of 4.95 out of 5. Since the user’s query was vague, the third option is a category called Pizza Recipes. You should recommend it if the user seems unsure. It was specifically curated to help as an inspiration of what types of tasks there are that are thematically related to the user’s search. If you recommend an option, you should also include a reason why you would recommend it. You should recommend the third option if the user seems unsure.  
 ### Input: User: *which one is your favorite?*  
 ### Response: Your response:

---

**Alpaca 7B:** My favorite is Search for Delicious by Maggie Hoffman. It has a rating of 4.  
**Guanaco:** I am not sure if I have a favorite, but I can tell you that the first one is a cookbook and the second one is a TV show.  
**Alpaca Lora:** I recommend the third option, Pizza Recipes. It was specifically curated to help as an inspiration of what types of tasks there are that are thematically related to the user’s search.  
**FLAN T5:** search for delicious

Figure 15: Example of a “hard” selection question, where the system needs to provide some opinion on which task would be the best for the user.

Step specific question (easy)

**Question:** *How long should I preheat the oven?*  
**Context:** Bake in the preheated oven for 30-35 minutes, or until a toothpick inserted into the center of the cakes comes out clean.  
**Task title:** chocolate layer cake

---

**ChatGPT:** The recipe does not specify a preheating time. You should generally preheat your oven for about 10 to 15 minutes before placing your chocolate layer cake inside. This gives the oven enough time to reach the desired baking temperature. If you're unsure, you can preheat your oven to the temperature specified in the recipe and then proceed with baking your cake. Keep an eye on the cake's progress and adjust the baking time as needed based on visual cues and the toothpick test mentioned in the context.  
**Guanaco:** 180 degrees Celsius for 10 minutes.  
**Alpaca 7B:** The oven is preheated for 35 minutes.  
**Alpaca Lora:** 30-35 minutes.  
**FLAN T5:** 30-35 minutes.

Figure 16: Example of an “easy” step specific question, where the answer can be extracted from the text. The question is *cursive*. The context each model has access to is shown in the top half, the answers are given below.

Domain question (easy)

**Question:** *What is the difference between cornstarch and cornflour*  
**Context:** In a bowl, combine cornstarch, lotion, and water. Note that, in the UK, cornstarch is typically called "cornflour" — the ingredients are identical.  
**Task title:** How to make your own face paint

---

**ChatGPT:** In the context of making your own face paint, the terms "cornstarch" and "cornflour" are used interchangeably, and their ingredients are identical. The primary difference lies in their names and the regional terminology. In the US and some other countries, the term "cornstarch" is commonly used, while in the UK, it is referred to as "cornflour." Despite the name variation, both terms refer to the same ingredient, which is a finely powdered, starchy substance derived from corn.  
**Guanaco:** Cornstarch and cornflour are both types of flour made from corn. They are both used as thickeners and are often used in baking and cooking. Cornstarch is a finer, lighter powder that is used to thicken sauces, soups, and puddings. It is also used as a binder in some cos...  
**Alpaca 7B:** Cornstarch and cornflour are both forms of starch made from corn. Cornstarch is more commonly used in the US, while cornflour is more commonly used in the UK. Cornstarch is a finer grind than cornflour and is used to thicken and stabilise sauces and soups. Cornflour is a coarser grind and is...  
**Alpaca Lora:** Cornstarch and cornflour are both types of flour made from corn. Cornstarch is a starch made from the endosperm of the corn kernel, while cornflour is a flour made from the entire corn kernel.  
**FLAN T5:** the ingredients are identical

Figure 17: Example of an “easy” domain question, where the answer can be extracted from the text. The question is *cursive*. The context each model has access to is shown in the top half, the answers are given below.

## C Additional Component Level Evaluation

Table 5: Neural Decision Parser action classification effectiveness by type.

Action	Precision	Recall	F1 Score
<i>answer_question</i>	0.761	0.913	0.830
<i>chit_chat</i>	0.667	0.343	0.453
<i>next</i>	0.833	0.833	0.833
<i>no</i>	0.750	1.000	0.857
<i>previous</i>	0.800	0.571	0.667
<i>repeat</i>	0.889	0.727	0.800
<i>restart</i>	1.000	0.833	0.909
<i>search</i>	0.889	0.814	0.850
<i>select</i>	0.953	0.953	0.953
<i>set_timer</i>	1.000	0.778	0.875
<i>show_more_details</i>	0.846	0.733	0.786
<i>show_more_results</i>	0.588	0.714	0.645
<i>show_requirements</i>	0.556	0.909	0.690
<i>start_task</i>	1.000	1.000	1.000
<i>step_select</i>	0.889	0.727	0.800
<i>stop</i>	1.000	0.625	0.769
<i>yes</i>	0.880	0.917	0.898

## D Conversation Rating analysis

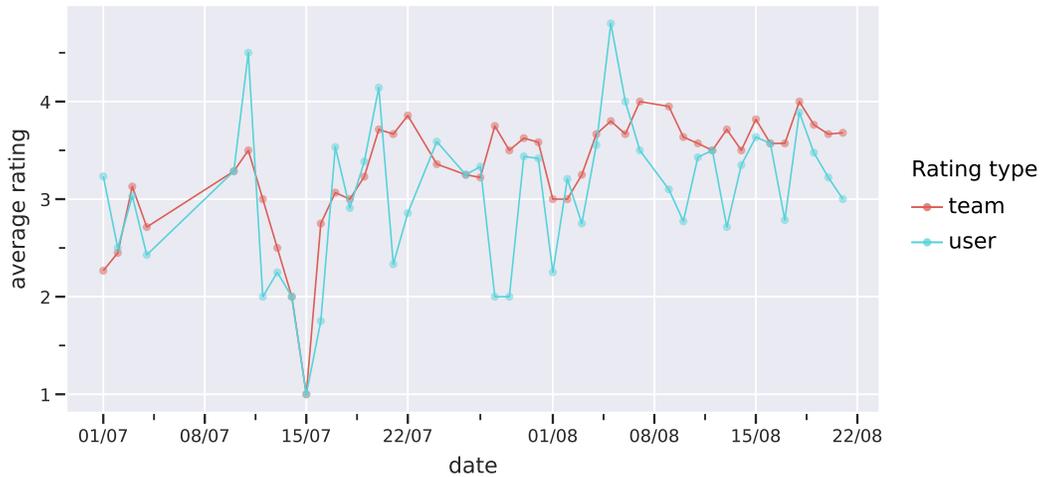


Figure 18: Comparing our annotated ratings vs user ratings. The conversations we annotated are a randomly selected subset of all user conversations occurring during the competition. We observe that whilst there is some overlap between user ratings and team ratings, user ratings do tend to randomly fluctuate.

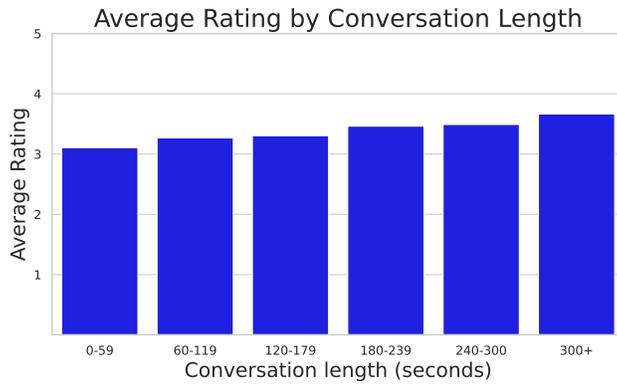


Figure 19: Comparing the user CSAT compared to the conversation length of the conversations over the months of July and August. We observe that GRILLBot-v2 receives better user rating the longer the conversation lasts.

Average rating difference between our rating and user rating

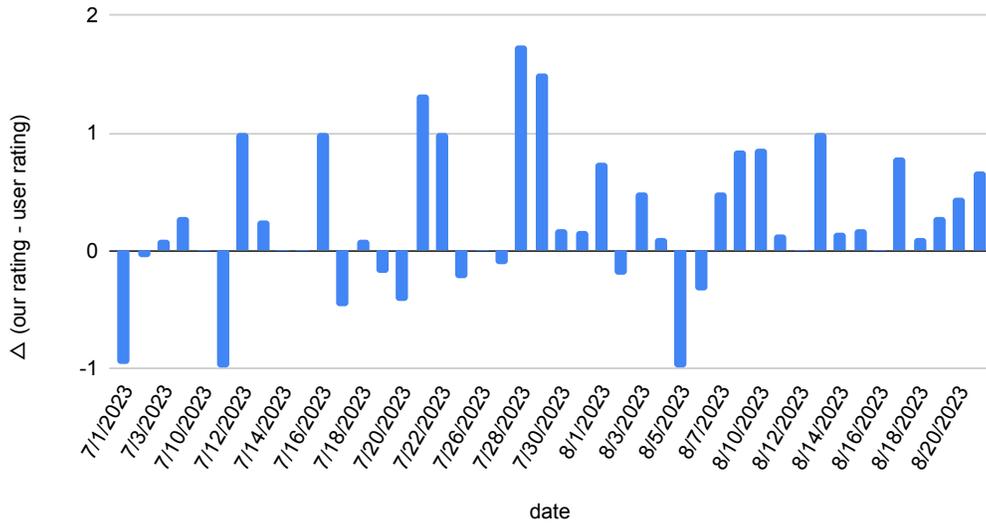


Figure 20: Comparing the average rating difference between our rating and the user rating. We observe that whilst there is an overlap, the team tends to rate conversations higher than users do.