Sage: A Multimodal Knowledge Graph-based Conversational Agent for Complex Task Guidance

Kaizhi Zheng* Jeshwanth Bheemanpally Bhrigu Garg Seongsil Heo
Dhananjay Sonawane Winson Chen Shree Vignesh S Xin (Eric) Wang

Abstract

This paper presents Sage, a task-oriented multimodal conversational agent developed for the Alexa Prize TaskBot Challenge 2. Focusing on cooking and DIY tasks, Sage integrates task-oriented dialogues with engaging general chats for a human-like interaction model. Its innovative hierarchical dialogue state management, based on hierarchical state machines, enables a flexible conversation flow managing both cross-task and inner-task intents. To offer comprehensive task-related insights, Sage employs a Multimodal Task Knowledge Graph, integrating diverse online data with advanced image generation and large language model techniques. Moreover, Sage pioneers an open-domain intent grounding approach with a T5-based model for high-level intent classification and an LLM-based model for open-domain demand understanding. These strategies allow Sage to handle complex user requests, fostering dynamic, relevant conversations. At the end of the semifinals, Sage achieved an average rating of 3.57/5.0.

1 Introduction

Artificial Intelligence (AI) has reshaped our lives in many ways, with one of its significant contributions being the rise of conversational agents. The Alexa Prize TaskBot Challenge 2 [1] provides an excellent platform for showcasing the potential of these agents, particularly in their ability to guide users through complex tasks. Among the innovative solutions presented at this competition is Sage, a task-oriented multimodal conversational agent that we introduce in this paper. Designed with the intention of emulating human-like interactions, Sage focuses on the domains of cooking and Do It Yourself (DIY) tasks. Its primary goal is to guide users through intricate tasks, while maintaining engaging conversations. The unique proposition of Sage lies in its seamless integration of task-oriented dialogues and engaging general chats, thereby making the interaction more dynamic and human-like.

To achieve this, Sage leverages novel approaches and technologies that challenge traditional task-oriented conversational models:

- **Hierarchical Dialogue State Management**: Sage breaks away from the conventional linear dialogues found in general chats, proposing a hierarchical dialogue state management method instead. This strategy, rooted in hierarchical state machines, enables a more flexible and structured conversation flow, effectively managing both cross-task (e.g., task searching) and inner-task (e.g., task step instruction) intents.
- **Multimodal Task Knowledge Graph**: To provide comprehensive guidance, Sage employs a Multimodal Task Knowledge Graph, a rich database of task-related information sourced from various online platforms. Enhanced with cutting-edge techniques in image generation and large language models, this knowledge graph supports user engagement and provides in-depth insights into the cooking and DIY tasks.

*contact address:kzheng31, jbheeman, bgarg, sheo1, dsonawan, wchen157, ss64293, xwang366@ucsc.edu

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

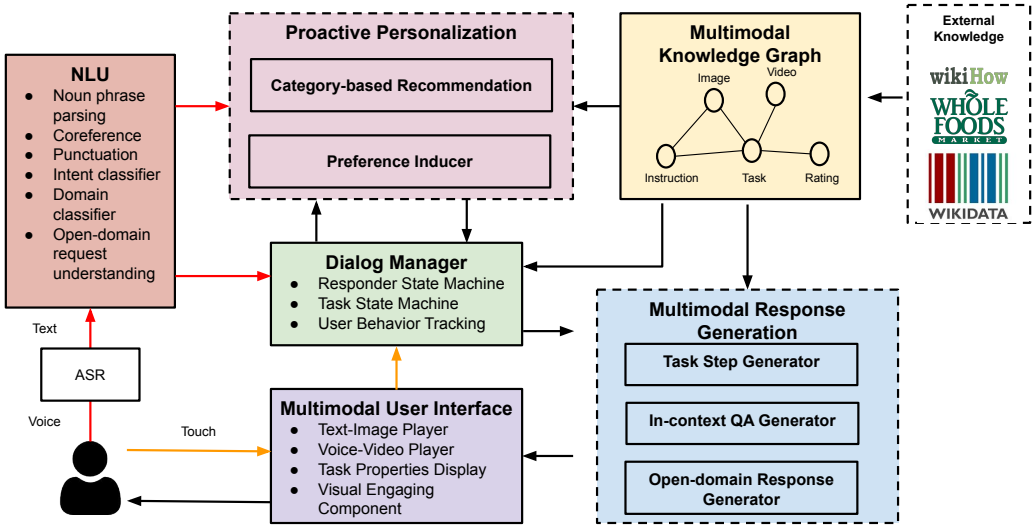


Figure 1: Overview structure of Sage system.

- **Open-Domain Intent Grounding**: To comprehend complex user requests, Sage employs a dual-model approach. It uses a T5-based model to classify high-level intents, such as search or question-answering, and an LLM-based model to understand open-domain demands (e.g., 'gluten-free recipes'). This enables Sage to accommodate a wide range of user requests and maintain dynamic, relevant conversations.

2 System Overview

2.1 Structure

To deliver robust, accurate, and engaging multimodal task guidance, we developed Sage, illustrated in Figure 1. This system encompasses six components: Natural Language Understanding (NLU), Proactive Personalization, Multimodal Knowledge Graph (MMKG), Multimodal Response Generation, Multimodal User Interface, and Dialog Manager. The first component, NLU, preprocesses user utterances by classifying intents and understanding open-domain requests. This is elaborated upon in Section 4. The second component, Proactive Personalization, hierarchically categorizes tasks under different keywords to guide users when they seek recommendations. This system enables users to identify suitable tasks by successively selecting relevant keywords. The third component, MMKG, aggregates task information from diverse external knowledge resources and augments it with cutting-edge techniques, discussed further in Section 3. Multimodal Response Generation, the fourth component, furnishes suitable responses when users inquire about task information or make out-of-domain requests. Detailed insights are provided in Section 4. Finally, the Dialog Manager and Multimodal User Interface form the remaining components of Sage, contributing to system management and user interaction, the details of which are expounded in the subsequent sections.

2.2 Hierarchical Dialog Manager

Given the complexity of task-oriented dialogue, which requires the integration of additional knowledge resources and necessitates navigation among different modules, traditional text-based dialogue management proves inadequate. To address these challenges, we developed a dialog manager rooted in a hierarchical state machine architecture, as shown in Figure 2. The top-level state machine orchestrates the transitions between various response generators, ensuring smooth and coherent dialogue flow. The second-level state machines, housed within each responder's node, facilitate the execution of specific actions these responders are designed for. This hierarchical structure not only

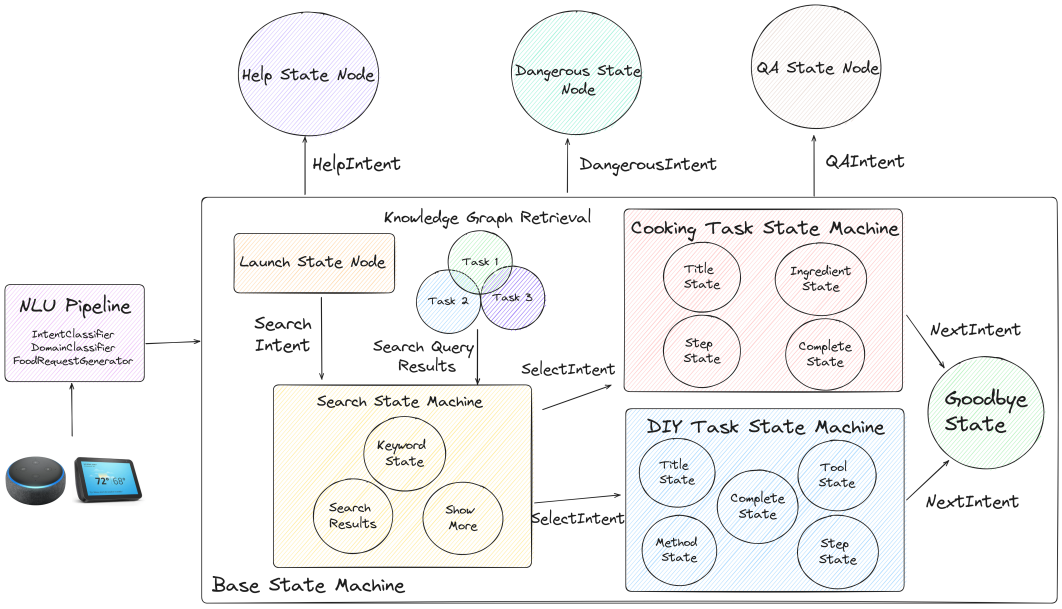
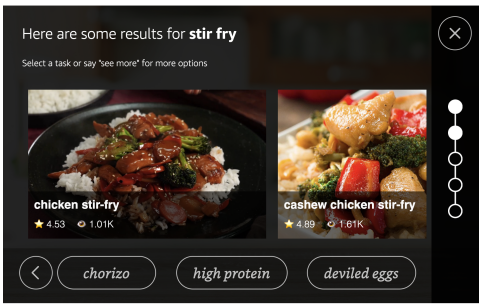


Figure 2: Hierarchical dialogue state management

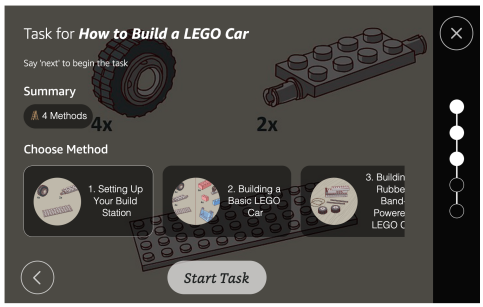
enables the handling of intricate user requests but also offers a high degree of adaptability, making it suitable for managing multi-modal, task-oriented dialogue.

2.3 Multimodal UI

We built a custom visual user interface to display to the user different tasks and their associated steps. We used emojis, step indicators, buttons, dynamic text, images, etc. as elements. We also designed our UI in a way to indirectly learn a user's interests for different tasks. For example, we use how long a user views a particular task when searching as a metric to create a user model for the user's interest.



(a) Task Search Screen



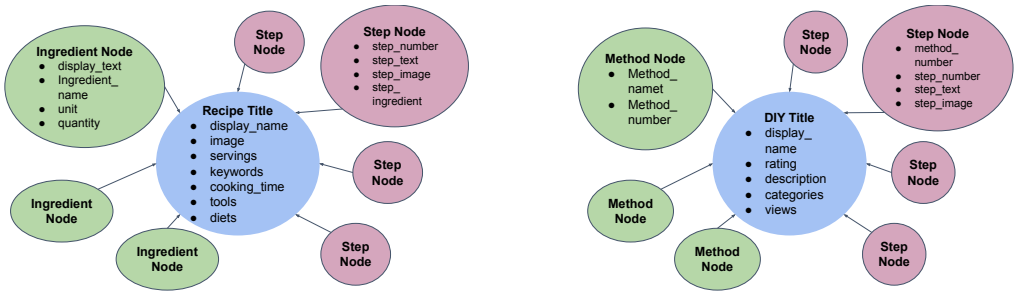
(b) Task Title Screen

Figure 3: Samples in multimodal UI

3 Knowledge Graph

3.1 Overview

We proposed an enriched data representation using a Neptune Knowledge Graph. Instead of just using provided datasets, we also scraped the recipe and DIY tasks from Wholefoods and WikiHow websites. The nodes and edges allowed easy data augmentation for ingredient images, recipe step images, fun facts, interesting task launch statements, and DIY tool names. The additional information is either added as a new attribute to an existing node or a new node. The graph structures for the recipe and DIY tasks are shown in Fig. 4, respectively.



(a) Recipe KG Structure (b) DIY KG Structure

Figure 4: Multimodal knowledge graph

3.2 Data Augmentation

3.2.1 Image Augmentation

Ingredient image In our endeavor to enhance the user experience, we recognized the challenge posed by the lack of user information, particularly in regard to their familiarity with the appearance of specific ingredients. We extended the multimodality of our bot by adding ingredient images for a recipe. While certain data sources conveniently provided us with comprehensive lists of ingredients, most others did not offer this essential information, necessitating an alternative approach to obtain the required data. It was a two-step process - extracting raw ingredient names from the ingredient text and generating ingredient images.

1 tablespoon minced garlic → minced garlic →

In the first step, we used Vicuna LLM[5] to generate raw ingredient names with few-shot inference. We scrapped ingredient text from Whole Foods, which has been mapped to the 83K unique raw ingredients. Later, in the second step, we generated images through the implementation of stable diffusion[14].

Step image A picture is worth a thousand words, and to truly empower our users in accomplishing their cooking goals and honing their culinary skills, we came up with the idea, to provide visual aids for every single step involved in the cooking process. With this vision in mind, we utilized stable diffusion techniques to generate high-resolution images that perfectly complement the cooking instructions. In this innovative approach, we directly fed the text associated with each cooking step as a prompt to the stable diffusion model. Through this method, we successfully created a vast collection of images, each serving as a visual guide to make cooking a delightful and seamless experience for our users. Figure 5a represents an image generated using the prompt "Put a large saute pan on the grill over low heat and add the basil butter. When the butter is hot add the corn kernels and saute for 3 to 4 minutes. Transfer to a serving bowl and garnish with torn basil leaves and scallions."

Tool Image In our DIY data processing endeavors, we encounter familiar challenges. The extraction of tools is extensively discussed in section 3.2.2 of our work. Leveraging the tool name as a prompt, we have successfully generated an impressive collection of over 36,000 unique tool images. An illustrative example of this is showcased in Figure 5b, where the prompt "paintbrush" resulted in the generation of a stunning tool image.

3.2.2 Text Augmentation

Task Summarization Task summarization is a fundamental component in enhancing user experience and task comprehension within a task-oriented chatbot. It plays a crucial role in ensuring users receive a coherent and comprehensive overview of the selected Cooking or Do-It-Yourself(DIY) task.

(a) Generated Step Image

(b) Generated Tool Image

Figure 5: Step and Tool Augmented Images

By providing a concise summary, users can quickly grasp the essential steps and goals of the task before undertaking it.

We used Large Language Models (LLMs) to generate offline summaries for each task in our knowledge base and update our knowledge graph. We used the task title, tools/ingredients required, and step instructions to prompt the LLMs to generate concise summaries. According to our human evaluations on different LLMs, from AlexaTM-20B [16], OpenAssistant-llama-30B [10] to LM-system's Vicuna-13B [5], with different prompting and decoding strategies, we found Vicuna-13B worked suitably well for task summarization. We post-processed the generated summary to eliminate bad and unfaithful generations based on n-gram overlap and bert-score of the generated summary and the task steps. In addition, we manually evaluated the generated summaries of 500 tasks (a diverse combination of cooking and DIY tasks) to see if they were coherent and grounded.

Fun Facts Generation Beyond providing functional assistance for cooking and DIY tasks, our chatbot aimed to engage users by incorporating intriguing and entertaining fun facts related to the selected task. We used LLMs to generate multiple quirky interesting titbits about the users' tasks.

A crucial aspect of fun fact generation involved enriching the knowledge base with a diverse range of interesting information related to cooking and DIY tasks. This entailed curating a vast repository of historical origins, notable pioneers, and amusing trivia associated with various tasks. We used wikidata as our primary data source. For each task, we crawled its nearest matching wiki article and mined it's parent and history section. This knowledge base acted as a valuable resource to draw upon when generating fun facts for users.

We prompted LLMs to generate multiple short fun facts for each task conditioned on the mined knowledge. We experimented with different LLMs, AlexaTM-20B [16], OpenAssistant-llama-30B [10] to LM-system's Vicuna-13B [5], with varied prompts and decoding strategies. Finally, we post-processed to filter out any toxic content generated. Additionally, we manually evaluated 300 diverse cooking and DIY tasks to evaluate the generated fact quality.

Tool Extraction A critical aspect of ensuring a seamless task experience is providing users with a comprehensive list of necessary tools for each task. Users find it beneficial to have a clear and concise list of tools necessary for a task. Such a list offers users a snapshot of the essential tools required, streamlining their preparation process and minimizing interruptions during the task execution. While some platforms, like Wholefoods, already provide ingredient lists for cooking tasks, other sources, such as WikiHow, lack explicit tool lists necessitating the need for tool extraction.

Our approach involved prompting LLMs to extract tools from step instructions of the task. We experimented with different LLMs, AlexaTM-20B [16], OpenAssistant-llama-30B [10] to LM-

Table 1: Example Text Augmentations.

Augmentation Strategies	Task	Augmented Text
Task Summarization	Soft Tacos	Savor the fusion of earthy re-fried beans, creamy avocado, and tender steak in every bite of these delectable soft tacos.
Task Summarization	How to email a resume	Send your resume to potential employers with confidence by following these simple steps for crafting a perfect email.
Fun Fact - Recipe	Blueberry Breakfast Trifle	The blueberry breakfast trifle is not only delicious but also nutritious! it contains antioxidants and anti-inflammatory properties that can help boost your immune system and reduce inflammation in the body, making them a great choice for a healthy breakfast or dessert option!
Fun Fact - DIY	How to Magnetize Steel	Did you know that magnetizing steel can help reduce the risk of injury or strain when performing heavy lifting tasks? by magnetizing a steel object, it becomes easier to lift and move due to its magnetic properties.
Tool Extraction	How to Use Tracing Paper	Extracted Tools: Picture, Tracing Paper, Tape, Graphite Pencil

system's Vicuna-13B [5], and decoding strategies. To ensure accurate tool extraction, the task instructions were split into individual sentences. This sentence-level granularity allowed the LLMs to focus on smaller units of text and pinpoint tools mentioned within each sentence. We use chain of thought prompting, with diverse instructions in prompts covering different categories, including examples of step instructions that don't require any tools. We then normalize the extracted tools and aggregate them across all the steps to get a list of tools for the task. This process is shown in Figure 6.

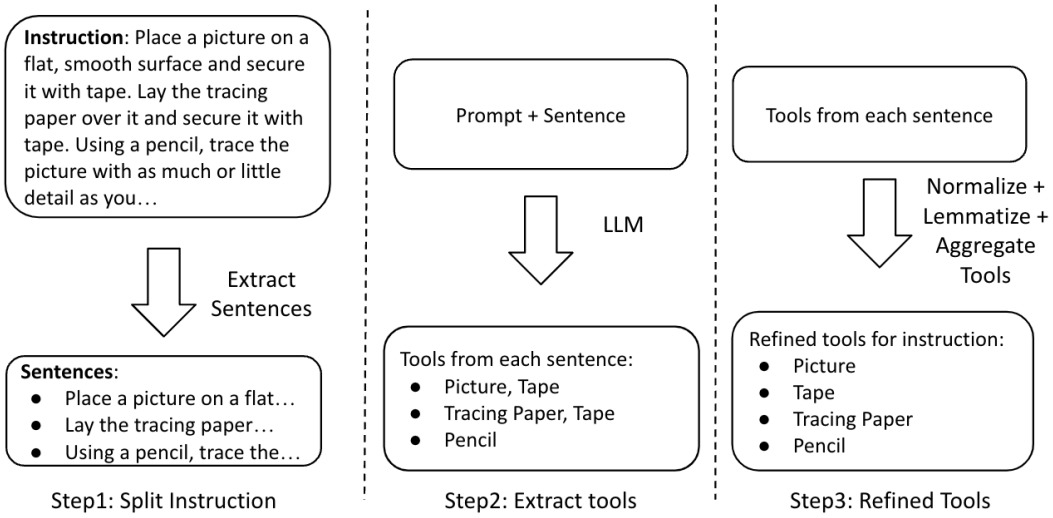


Figure 6: Tool Extraction Process

While this prompting method proved effective in extracting tools, certain challenges, such as bad tools being extracted (false positives), needed mitigation to improve the overall reliability.

To achieve this, we implemented the following strategies:

- To enhance robustness and reduce false positives, we prompted the LLMs multiple times at different levels of sentence granularity. This involved providing prompts with variations in the number of sentences included, such as taking one sentence at a time or combining 2-3

sentences with overlapping strides. By doing so, we obtained multiple tool sets corresponding to different levels of context, ultimately reducing the likelihood of extracting incorrect or irrelevant tools.

- To refine the tool extraction output, we took the intersection of the tool sets generated from the multiple prompting levels. By considering only the tools that consistently appeared across different prompting levels, we improved the precision of the extracted tool list, minimizing the inclusion of erroneous tools.

It is essential to acknowledge that while the intersection of multiple tool sets enhances precision by reducing false positives, it slightly reduces recall. By prioritizing precision, we aimed to ensure that the tools provided to users were accurate and relevant. The slight trade-off in recall was deemed acceptable, as it allowed us to maintain the quality of the tool list without compromising on its correctness.

4 Neural Module

4.1 Intent classifier

A critical component in our NLU pipeline is being able to detect user intents, especially search and select intents. The first approach was a encoder model, using BERT, with a classification head to predict the user's intent given the current user utterance along with the previous system utterance. The model also jointly predicts the domain the user is referring to, whether that be cooking, DIY, or general in a multitask setting [12]. The predicted intents are search, jump, previous, next, more, question, repeat, action, chat, no, stop, select, yes, and other. We create our dataset using the previous year's Taskbot Challenge 1 past conversation dataset shared by Amazon [9]. In addition, we use the wizard of tasks dataset provided by Amazon as support for the intents [6]. The second approach used a T5 generative encoder-decoder model where we generated the above intents, given the user utterance and context. The results for these models are shown below.

Table 2: Intent Classification F1 Scores

Intent Model	Dev	Test
BERT-base	0.671	0.653
T5-base	0.742	0.728
T5-large	0.765	0.754

4.2 Open-domain Request Understanding

Understanding open-domain requests is critical in the realm of task-oriented dialogue systems. For instance, a user may request, "I want a new year recipe with beef." In this query, two specific search parameters are evident - the occasion is 'new year,' and the recipe must include beef. To effectively address such scenarios, we've fine-tuned a language model (pythia-1.3B) [4] to translate utterances into JSON-format requests, a process detailed in Figure 7. This model, referred to as the Open-Domain Request Understanding model, is proficient at deciphering nuanced search requests. For model training, we first identify all possible values for several keys, such as "occasion", "dietaryFilters", "ingredients", etc., within our collected database. We then generate a random assortment of these key combinations and prompt a pretrained large language model [5] to simulate corresponding user text requests. This simulation process enables the creation of diverse data pairs, which are then used to fine-tune our language model for optimal performance.

4.3 General LLM Responder

One of the neural modules in our system is dedicated to a general-purpose Large Language Model Neural Response Generator (NRG). Our goal is to cater to users who may have intents beyond the domains of cooking, DIY, or other general-purpose multitasking settings. To address this, we have implemented a chitchat module that can handle various general-purpose questions about cooking, DIY, and other activities, providing users with a comprehensive chatting experience and expanding their knowledge about different tasks.

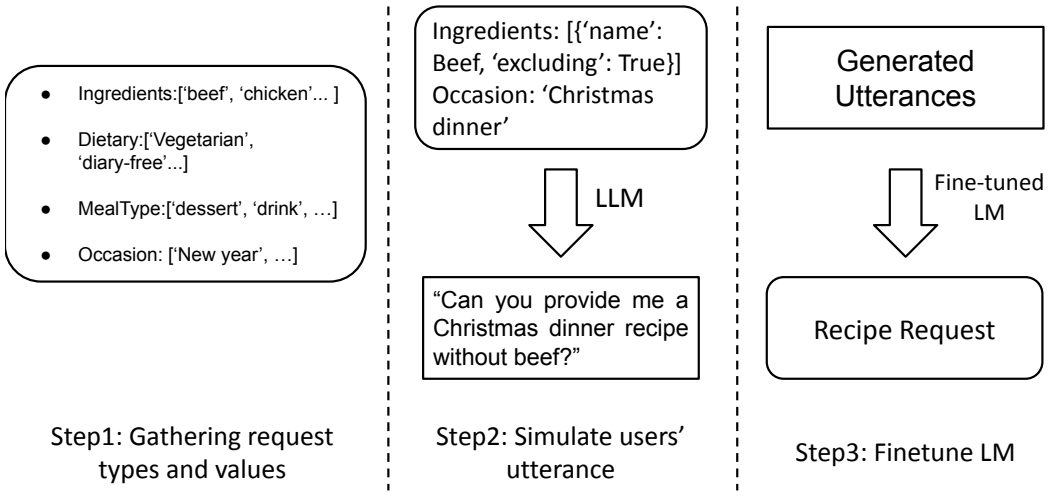


Figure 7: Open-domain Request Understanding Process

For this purpose, we adopted the fastchat-t5-3b-v1.0 [17] model as a zero-shot NRG. This NRG is designed to process user utterances along with the chat histories of the bot, enabling it to excel in question-answering scenarios. To evaluate the effectiveness of our approach, we conducted experiments using the WikiHow dataset. Specifically, we simulated user utterances using the Community Q&A sections of WikiHow, which consist of questions asked by users, along with crowd-sourced responses.

During the evaluation phase, we employed 1000 samples from the WikiHow question-answer dataset and compared the performance across three different settings: Question, Question w/ Title, and Question w/ Title & Context. To assess the model's capabilities, we utilized three widely adopted NLP metrics, namely SBERT[13], ROUGE[11], and METEOR[3].

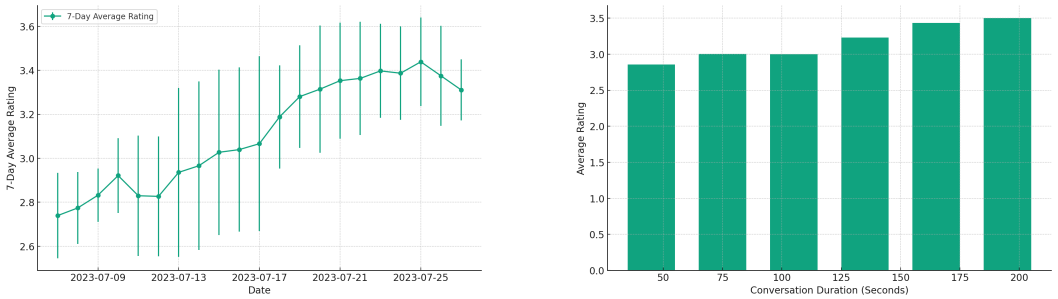
The ultimate objective of these experiments is to enhance user experience by ensuring that the NRG can effectively handle a wide range of questions and deliver informative responses across various contexts. Through this evaluation process (Table 3), we aim to achieve a high level of performance and accuracy, thereby providing users with valuable and engaging interactions.

Table 3: Experimental result on different prompt and the average of each score with 1000 samples.

Fastchat-T5	SBERT	ROUGE-L	METEOR
Question Only	0.5500	0.1608	0.1585
Question w/ Title	0.5574	0.1655	0.1636
Question w/ Title & Context	0.5712	0.1716	0.1672

5 Analysis

In the present study, the performance of the system was evaluated by examining user interactions and assessing the quality of ratings. In Figure 8a, the line graph represents the change in the 7-day average ratings over time, with error bars indicating the standard deviation. The standard deviation measures the variance of the ratings. The figure shows that the 7-day average rating generally increases throughout the month, and the ratings' variance progressively decreases, indicating that Sage has steady progress and is robust for ratings. By the end of 07/28, our last 7-day average rating achieved 3.34/5.0. From Figure 8b, we find that the users with high conversation duration are more willing to leave with higher ratings, which is consistent with expectations.



(a) 7-day average ratings over time

(b) Average ratings based on conversation duration

Figure 8: Performance by the end of the semifinal.

6 Ongoing Research

Preference Inducer Preference inducer techniques are crucial in tailoring the bot's services to individual users and meeting their specific needs, resulting in a highly satisfying user experience and enhanced trustworthiness. In our research, we collect user conversations from DynamoDB to construct user profiles and use a BERT model to derive meaningful embeddings. Furthermore, we curate high-frequency keywords from past logs and measure their similarity to user profiles, thereby selecting the top 3 relevant words for personalized recommendations. This approach enables us to offer highly tailored topics surpassing the limitations of the random keyword-based approach.

However, our research tackles several challenges in effectively utilizing personalized information. Incorporating time information is crucial, as user interests dynamically change over time[2, 8]. Moreover, filtering out irrelevant conversations and prioritizing essential ones becomes imperative [7]. For instance, dialogues like "Alexa, hello. How are you?" provide negligible information. It fortifies our preference inducer, resulting in a more focused and streamlined experience, ultimately elevating overall user satisfaction.

Video Generation Text-to-video generation offers superior advantages over text-to-image generation by conveying a broader range of visual content through dynamic sequences of frames. By representing real-world scenarios and delivering impactful narratives, text-to-video generation elevates the creation of multimedia content, fostering more captivating and immersive user experiences.

A cutting-edge text-to-video(T2V) model, known as the spatiotemporal factorized diffusion model, has emerged recently [15]. It exploits a joint text-image prior, eliminating the dependency on paired text-video data for training. Additionally, the model incorporates sophisticated super-resolution states to enhance the quality of generated videos. We are aiming to apply T2V model to provide users with engaging and dynamic video content, enabling them to experiencing a more immersive and captivating viewing experience.

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. Alexa, let's work together: Introducing the second alexa prize taskbot challenge. In *Alexa Prize TaskBot Challenge 2 Proceedings*, 2023.

[2] Reham Alabduljabbar, Manal Alshareef, and Nada Alshareef. Time-aware recommender systems: A comprehensive survey and quantitative assessment of literature. *IEEE Access*, 2023.

[3] Satanjeev Banerjee and Alon Lavie. 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, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[4] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023.

[5] 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. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[6] Jason Ingyu Choi, Saar Kuzi, Nikhita Vedula, Jie Zhao, Giuseppe Castellucci, Marcus Collins, Shervin Malmasi, Oleg Rokhlenko, and Eugene Agichtein. Wizard of tasks: A novel conversational dataset for solving real-world tasks in conversational settings. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3514–3529, 2022.

[7] Yang Deng, Wenxuan Zhang, Weiwen Xu, Wenqiang Lei, Tat-Seng Chua, and Wai Lam. A unified multi-task learning framework for multi-goal conversational recommender systems. *ACM Transactions on Information Systems*, 41(3):1–25, 2023.

[8] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. Time-sensitive recommendation from recurrent user activities. *Advances in neural information processing systems*, 28, 2015.

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

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

[11] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[12] Pengfei Liu, Kun Li, and Helen Meng. Out-of-scope domain and intent classification through hierarchical joint modeling. *arXiv preprint arXiv:2104.14781*, 2021.

[13] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[14] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn 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.

[15] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[16] Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, Chandana Satya Prakash, Mukund Sridhar, Fabian Triefenbach, Apurv Verma, Gokhan Tur, and Prem Natarajan. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model, 2022.

[17] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.