TaskMaster: A Novel Cross-platform Task-based Spoken Dialogue System for Human-Robot Interaction

Carl Strathearn
Edinburgh Napier University
School of Computing, Engineering and the Built Environment
c.strathearn@napier.ac.uk

Yanchao Yu
Edinburgh Napier University
School of Computing, Engineering and the Built Environment
y.yu@napier.ac.uk

Dimitra Gkatzia
Edinburgh Napier University
School of Computing, Engineering and the Built Environment
d.gkatzia@napier.ac.uk

ABSTRACT

The most effective way of communication between humans and robots is through natural language communication. However, there are many challenges to overcome before robots can effectively converse in order to collaborate and work together with humans. This paper introduces TaskMaster [1] a novel cross-platform spoken dialogue system (SDS) for human-robot interaction (HRI) which employs a neural language model to generate responses in the context of task-based situations. In contrast to previous works that have employed templates and canned text for dialogue in HRI, we show that the dialogue output of TaskMaster is more flexible than a template-based variation. In a series of task-orientated case studies and a video demonstration[2], we show that in real-world settings TaskMaster can generate more relevant responses to questions, identify missing objects and offer alternatives, confirm and clarify aspects of a task, and adapt to unpredictable situations more effectively than traditional template approaches used in HRI.

KEYWORDS

Spoken language interaction, Task-based dialogue, Human-robot interaction, Natural language generation

ACM Reference Format:
Carl Strathearn, Yanchao Yu, and Dimitra Gkatzia. 2023. TaskMaster: A Novel Cross-platform Task-based Spoken Dialogue System for Human-Robot Interaction. In *Proceedings of The Joint CUI and HRI Workshop at HRI 2023 (Conference acronym 'HRCI23).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Increasingly, people expect social robots to interact fluently in real-world settings, using natural language to help them achieve goals or complete tasks [15]. For example, Astro [5], Aido [27] and Kuri [21] are social companions and domestic assistants. One key feature of such interactions is that dialogue requires flexibility, as goals might change dynamically with the progress of the task at hand [13]. However, the adoption of current SDS in HRI is hindered by the restrictiveness of platform-dependent software. Currently, platform robots used in domestic and social settings such as Pepper [23], Nao [29] and Furhat [4] have proprietary software that can only be customised by adding or adapting 'skills'. Recent studies [17], [16], [40] have developed custom software to work with the existing components of the above-mentioned robots because of such limitations. However, as these systems are designed to work with specific robots and their existing components, reduce their deployability to other robots as-is. In addition, dialogue and response generation have not received much attention as a problem of focus in the HRI community outside of venues such as NLG4HRI [9] and NLiHRI [1]. Instead, hand-crafted templates for response generation are the most commonplace means of communication for robots. Templates however are often bespoke, monotonous and thus inflexible, hard to generalise or porter to a new domain, and labour-intensive with restricted turn-taking and knowledge transference [39]. Therefore, it is critical to develop platform-independent SDS that can effectively manage unscripted dialogue across non-commercial robots to encourage greater study and application in dialogue and response generation. Such systematic limitations have long underpinned the use of non-computational methods of spoken language interaction in HRI, such as the wizard of oz (WOZ), telepresence and costume [30] [26] [41]. The advantage of non-computational methods is that the human operator can effectively handle impromptu and complex dialogue states through a robot in real-time, in real-world environments [25]. However, the contributions of such methods to HRI are limited to areas such as dialogue modelling and user interaction studies that do not require automation [12]. The Amazon Taskbot challenges [14] recently explored some areas of flexible dialogue management in the cooking domain, showcasing chatbots such as GRILLBot [22] Tartan [36], Howdy Y'all [38] and Miutsu [37]. However, the challenges assume precise conditions and manage missing or incomplete user knowledge, outside of the written/spoken dialogue. For example, the users have access to all necessary components to complete the task and can locate objects hidden from sight within a given environment. Finally, it is assumed that users know what specific objects look like, their utility and how to handle them in order to perform tasks, which might not be generalisable to other domains. Such assumptions of knowledge and environment do not address real-world human factors central to HRI, which may require other types of domain knowledge to resolve issues. Here we present TaskMaster, a platform-independent task-based SDS for HRI, where a robot takes the role of an information giver (IG) and a human assumes the role of the information follower (IF) with a joint goal to complete a task, such as cooking. Specifically, the IG

---

[1] https://anonymous.4open.science/r/CiViL-44C2/README.md
[2] https://youtu.be/Jkubvfwuqec

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'HRCI23, 13–03, 2023, Stockholm, SWE
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

has access to a document that describes a task such as a recipe, has knowledge about cooking ingredients and substitutes as well as how to use cooking tools, what they look like, and their common places of storage. The IF is able to request a specific recipe and receive instructions, but at the same time has the opportunity to request clarification or ask for substitutes, or ways to perform a sub-task. This type of dialogue management requires flexibility as the goal of the communication can be briefly altered from cooking a recipe to requesting information on how to use a tool, and then resuming to the main overall goal. Such phenomena are commonplace in everyday dialogue between humans, for instance at workplaces where people work towards a common goal, or when people follow instructions to complete tasks, such as assembling furniture. Our novel system can be easily adapted to similar applications in instructional domains or solve similar tasks that require changes in the communication goal in hands-busy scenarios without having to switch between objects such as written manuals or scrolling on screens.

The contributions of this paper are as follows:

- We present a platform-independent SDS for task-based dialogue in HRI.
- Modular system design that can be adapted for new tasks in different domains and integrated with new modalities such as object recognition and user tracking.
- We present a bespoke response generation module that can be trained for any real-world task.
- We make the system available as open-source.

2 TASKMASTER

In this section, we provide an overview of the TaskMaster system architecture; 2.1 dataset and knowledge bases 2.2; dialogue management; 2.3 system installation and deployment; 2.4 with a practical example of deployment in a real robot.

2.1 System Architecture

The TaskMaster system architecture is shown in Figure 1, composed of the following modules:

- *Automatic Speech Recognition (ASR)* is used to transcribe user utterances from speech to text. We are using the Google cloud platform (GCP)[3] for ASR.
- *Natural Language Understanding (NLU)* which is responsible for creating semantic representations of the input text. We are using the existing Rasa NLU pipeline[4] fine-tuned on our own dataset.
- The text representations are then passed to the *Dialogue Manager* which selects the dialogue acts based on the input representation and the external knowledge database, further discussed in Section 2.3.
- An external *knowledge base* stores domain knowledge related to the task at hand (see Section 2.2).
- The *Natural Language Generation (NLG)* module is responsible for generating responses based on the selected dialogue acts. (see Section 2.3)
- The generated response is then passed to the *Text-to-Speech (TTS)* module and turned into spoken utterance, using GCP outputted to a robot, as described in Section 2.4.

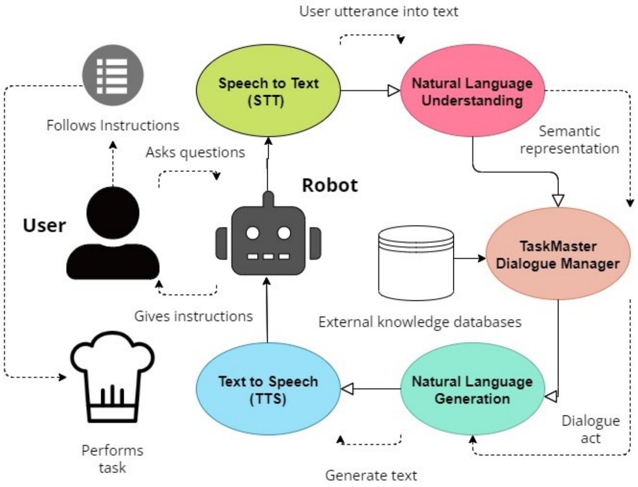

Figure 1: TaskMaster system architecture

2.2 Dataset and Knowledge Bases

As a case study for TaskMaster, we use the Task2Dial dataset[5] [33], a corpus of over 350 instructional conversations between an IG and IF in the cooking domain, i.e. the IG has knowledge of a cooking recipe and provides instructions to the IF so that the IF can complete the recipe step by step. Our system handles two types of knowledge: knowledge modelled in dialogue and external databases.

Knowledge modelled in dialogue: Our system is trained on dialogues between an information giver and a follower. The dialogues are focused on five commonsense knowledge types (temporal, spatial, contextual, behavioural and object knowledge). For example, a recipe may state, "peel the potatoes", which assumes that the user has knowledge of what tool is needed, correct handling, common storage, and what it looks like, however, the dialogues don’t make such assumptions and instead, the corresponding instruction would be: ‘using a vegetable peeler, peel the potatoes. To reduce the need for a large number of examples in the dialogue, i.e modelling all possible domain knowledge within the instructions, we created a series of external knowledge bases that contain different types of domain knowledge. This allows us to contextualise and extract knowledge from dialogue within these databases at any stage of the task, rather than modelled within specific dialogue scenes.

External commonsense knowledge bases: These are CSV files that hold information about ingredients and objects, and hold long descriptions about the utility, appearance, handling, similarity to other objects, and common storage locations. In the ingredients database, we include a list of alternative ingredients, these ingredients are swappable as is and do not require changes to the task.

---

[3] https://cloud.google.com/

[4] https://rasa.com/docs/rasa/tuning-your-model/

[5] https://huggingface.co/datasets/cstrathe435/Task2Dial

2.3 Dialogue Management

The TaskMaster dialogue manager (DM) is built on the RASA 2.0 [6] environment for; training, encoding actions and intents, rules and state tracking. Integrated into this environment are controls for sequencing scripted data and generating dialogue. Dialogue is generated from a text file that contains all the IG dialogues from the original recipes and data from the external knowledge databases. The TaskMaster user interface (UI) is a standard terminal interface and can be used in most python-enabled integrated development environments (IDE), such as PyCharm [6], Google Colab [7] or IDLE [8].

Natural Language Generation (NLG) is the process of generating fluent and natural responses from underlying non-lexical representations[11] in the context of dialogue. TaskMaster uses a hybrid approach to NLG. Neural NLG is used for question-answering based on the confidence threshold, whereas a limited number of templates are used for greetings. By default, TaskMaster uses a version of BERT [10] proposed by RSVP.ai[9] which is available through Huggingface [10] for response generation. As TaskMaster is modular, this module could be easily replaced with a different language generation model such as T5 [24], RoBERTA [19] or GPT-3 [8] if the task requires it. This adaptability is particularly important when training a new model in a different domain as alternative models may generate more accurate and concise responses depending on the task at hand. Here, BERT was selected after empirical investigations with the types of commonsense knowledge we wanted to capture. To summarise, we use a rule-based model (Rasa NLU) for the generation of recipe instructions, as typical in SDS. However, the flexibility of the dialogue comes from the DM: we capture questions that are out of the recipe plan (e.g. ingredient swap) and instead we use a foundation model [7] to answer these questions based on the different types of external commonsense knowledge databases. The NLU module informs the DM's actions through the usage of case statements (e.g. ingredient-swap, tool usage, specific recipe, all recipes) to move outside and within paths, this allows us to perform subtasks before returning to the main overall goal of the task. Putting all these processes together allows us to manage "surprising" scenarios and enable natural dialogue.

2.4 Installation

A dockerised version of the project is provided in the project's GitHub repository to facilitate quick installation of the system on a local machine [11]. TaskMaster is coded in Python so it can be adapted and deployed on most IDEs. The system can be run on various operating systems such as Windows, OSX and Linux to increase system compatibility for robotic systems. Our system is open-source and platform-independent as it can be deployed, outputted and changed in any number of ways on different platforms for different tasks. It is important to clarify that we did not build our system without consideration or beside a robotic system, we built TaskMaster without consideration of a 'specific robotic system design'. This is a significant and ongoing problem in bespoke robotic designs that do not fit into conventional system structures. For example, the open-source multi-functional HRI system 'HARMONI' [31], for rapid robotic system deployment, focuses on synthesising specific sensory modalities (speech, vision and control), yet, not all bespoke robotic systems are configured this way, and it is unclear how removing one or more modalities may affect the functionality of the other/s. However, central to any social/assistive robot is an SDS, which can then be enhanced with other modalities, as is, such as computer vision or other forms of control, i.e gesturing, in a non-specific way. Thus, we designed TaskMaster as an SDS for 'task-based' flexible dialogue and response generation, rather than focusing on specific or generalised designs and configurations in robotics.

3 TASKMASTER ROBOT DEPLOYMENT

The TaskMaster dialogue manager uses the standard audio ports found on most computers, tablets and bespoke or commercial robots. Audio data is inputted through the microphone port and outputs through the headphone port using 3.5mm stereo cables to enable automatic speech recognition (ASR) for the dialogue system. The audio cables are inserted directly into the standard onboard speaker and microphone system of a robot for use as is. However, audio can also be outputted using a Bluetooth speaker and microphone to extend the range for mobile robotic systems. Specific libraries can be downloaded to use audio devices that are 'non-standard' which can be installed on a computer and activated through the Python framework to enable audio input and output. The ASR components can also be directly inputted into microprocessors used in robotics such as Arduino and Raspberry Pi, this can be used to augment spoken dialogue for other modalities such as lip synchronisation and audio-directed user tracking. We provide a case study to demonstrate how TaskMaster can be deployed on a bespoke robotic system called 'Euclid', using the robots embedded speakers and microphone for ASR and parsing the audio data through an onboard Arduino microprocessor to enable lip synchronisation [35]. In a video demonstration [12], we show how TaskMaster can be run simultaneously with externally embedded robotic control systems using a Pololu microprocessor [20] to control facial expressions and a Kinect camera sensor for user tracking [32] for a multi-modal system approach.

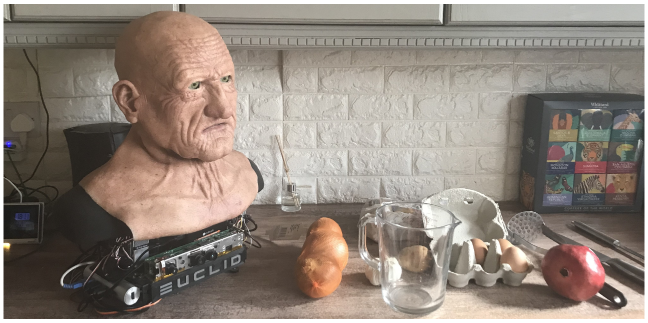

Figure 2: Robot deployment example using Euclid

---

[6] https://www.jetbrains.com/pycharm
[7] https://colab.research.google.com/
[8] https://docs.python.org/3/library/idle.html
[9] https://rsvp.ai/cn
[10] https://huggingface.co/rsvp-ai/bertserini-bert-large-squad
[11] https://anonymous.4open.science/r/CiViL-44C2
[12] https://youtu.be/Jkubvfwuqec

4 SYSTEM ANALYSIS

In this section, we compare and analyse the dialogue output of TaskMaster against a rule-based dialogue agent, called ChefBot [34], using the example template provided with RASA open source [13]. They are both integrated with the same Rasa NLU model. Both systems were installed in a humanoid robot named Euclid (as seen in Figure 2) and tested in a practical task-based setting using cooking recipes from the Task2Dial dataset. As Chefbot previously achieved good results on our cooking task [34] but was limited in some aspects, i.e flexible dialogue and response generation, we explored these limitations in the development of TaskMaster. It is also important to clarify that we are not focused on comparing existing systems, but rather methods. Hence, we selected a common method in robotic DM’s using a rule-based model. However, it is important to note that our system is flexible so other developers/researchers can swap a generation model with a more suitable one in their domain.

4.1 Task-based Challenges

In this section, we explore five task-oriented challenges in the cooking domain. In our cooking task, the IG has access to the recipe and gives instructions to the IF who performs the task. However, in real-world HRI scenarios, conditions may change or additional information may be needed for the IF to better understand the task. Therefore, to better demonstrate TaskMaster’s functionality, we consider the following scenarios.

4.2 Relevant and Irrelevant Domain Knowledge

In the context of SDS design [28], relevant domain knowledge is information that is directly related to a user’s question and irrelevant knowledge is superfluous information, such as additional dialogue that may be part of a larger data string. In questions 1-3 the user asks questions on different aspects of the task that are not directly grounded in the data. i.e. what does an *object* look like?. For this, we sub-categorise relevant object knowledge, such as the ability to visually describe the appearance of an object, an object’s common storage location and an object’s utility. The objective of this scenario is to analyse what information is relevant and irrelevant when the user asks a specific question relating to objects used in the task, as shown in Table 1. Furthermore, in SDS, it is easier to understand shorter dialogue rather than paragraphs as it can be more difficult to retain or locate specific details in long-spoken sentences.

Analysis. We see that Chefbot only captures ’chopsticks’ as an utterance ’slot’ and gives the same full response for all questions on a specific object from the utensils dataset. However, TaskMaster is able to generate more relevant knowledge to user questions from context, thus accurately shortening the response to a few short words. This allows greater scope for asking specific questions in different ways to capture different types of knowledge. This may be particularly valuable in HRI scenarios as it bypasses the rigidity of intent modelling in rule-based systems i.e. modelling different ways people may ask a specific question for more natural and less generalised QA.

TaskMaster	**ChefBot**
Question. 1 **Recipe 8** U(1): **What are chopsticks?** S(1): a pair of slender plastic or wooden sticks held between thumb and fingers	Question. 1 **Recipe 8** U(1): **What are chopsticks?** S(1): Chopsticks are a pair of slender plastic or wooden sticks held between the thumb and fingers, they are usually stored in a kitchen drawer.
Question.2 **Recipe 8** U(2): **Where in the kitchen are chopsticks usually stored?** S(2): a kitchen drawer	Question. 2 **Recipe 8** U(2): **Where in the kitchen are chopsticks usually stored?** S(2): Chopsticks are a pair of slender plastic or wooden sticks held between the thumb and fingers, they are usually stored in a kitchen drawer.
Question. 3 **Recipe 8** U(3): **What do chopsticks look like?** S(3): slender plastic or wooden sticks	Question. 3 **Recipe 8** U(3): **What do chopsticks look like?** S(3): Chopsticks are a pair of slender plastic or wooden sticks held between thumb and fingers they are usually stored in a kitchen drawer.

Table 1: Example of Relevant Knowledge QA

4.3 Flexible Question Answering (QA)

Rule-based models learn rules that define how an SDS responds using slot filling [42] without allowing variability or flexibility in responses. On the other hand, generative models generate responses without using pre-defined templates [18], but also without working towards achieving a communication goal, i.e., dialogues are mostly chit-chat. In the context of our cooking task, it is vital to ensure the user has followed, understood and completed a specific step of a given recipe before moving to the next step. i.e. Do I need to complete objectives (A) and (B), before I go to the next step? As such questions may be asked at any stage of the task, it may require moving outside of the sequences of a recipe and momentarily having a parallel sub-dialogue about object utility or ingredient substitution. In questions 4-6, we consider alternative objects and missing objects in cooking scenarios as part of flexible QA. For example, the user may not have a specific utensil or cooking equipment as stated in a recipe. However, drawing on additional data, the IG may be able to suggest alternative utensils for the underlying context e.g. ’Use a gas stove instead of an electric stove’. Similarly, a recipe may refer to an action without defining an object, e.g. ’peel the potatoes’, here the user is expected to know what tools are needed for the task. Yet, in real-world scenarios, the user may not know what tools they need, here we explore how missing object knowledge in dialogue can be generated from other data sources, in order to complete a task. These concepts are significant in task-based HRI where the IG needs other domain knowledge for the IF to complete a task, due to incomplete information or missing objects, as shown in Table 2.

Analysis. A limitation of both TaskMaster and ChefBot comes from the NLU’s incapability to interpret questions that contain intents, but are not answerable by the corresponding intent and dialogue act, as shown in the results of Question 4. For example, ’Do I need to preheat the oven for the intent: next part of the recipe’. This is problematic to interpret as ’next’ is an intent that moves the conversation onto the next step in the sequence. However, moving to the next step does not answer the user’s question. We aim to further explore this issue by further training the NLU to recognise such intents.

---

[13] https://github.com/RasaHQ/rasa

TaskMaster	ChefBot
Question. 4 **Recipe 13** U(1): **Do I remove the Bundt from the oven before I do the next step?** S(1): Now preheat the oven at 180°. Using your hand, take 1-2 tablespoons of butter and grease a large Bundt. When it has been greased, take 1 teaspoon of flour and dust it over the butter.	Question. 4 **Recipe 13** U(1): **Do I remove the Bundt from the oven before I do the next step?** S(1): To make the icing, start by using a colander to sift 6 tablespoons of confection s sugar into a small bowl
Question. 5 **Recipe 1** U(1): **I don’t have an electric griddle what can I use instead?** S(1): flat top grill.	Question. 5 **Recipe 1** U(1): **I don’t have an electric griddle what can I use instead?** S(1): Please repeat the question.
Question. 6 **Recipe 212** U(1): **What can I use to roll the dough?** S(1): A rolling pin.	Question. 6 **Recipe 212** U(1): **What can I use to roll the dough?** S(1): You then have to place each of the slices onto a flattened out paper cupcake case on a baking tray, or into a greased muffin tin.

Table 2: Example of Flexible Question Answering (QA)

In Questions 5 and 6, we examined how the systems manage questions on aspects of the task that are incomplete i.e. requiring alternative methods. In Question 5, TaskMaster is able to suggest a flat-top grill instead of an electric griddle, this data is cross-referenced with the data in the external database for object knowledge to answer the question. In contrast, ChefBot was unable to understand the question and generated ’Please repeat the question’. In Question 6, we examined how the systems would handle missing information in the original dialogue. TaskMaster was able to generate a correct response for a missing object in the original dialogue by understanding the context of the task, i.e. rolling the dough, demonstrating flexibility between understanding the method (A) and tool (B). However, ChefBot was unable to link the question to the object knowledge database as the action Rolling pin was not mentioned in the utterance, in turn, Chefbot moved to the next step in the dialogue sequence to continue the conversation.

4.4 Confirmation Questions

Asking for confirmation in task-oriented dialogue typically limits responses to pre-defined questions [2], e.g. ”is it correct that you want to book a taxi? please say ’yes’ or ’no’.” However, in a cooking scenario confirmation questions may be more complex, for instance, ”can you tell me how many (specific ingredient) I need”? or ”how long do I need to cook these (ingredients) for?”. In questions 7-9 we analyse how confirmation questions are managed (as shown in Table 3), i.e. does the user need to go through previous steps or ask for the instruction to be repeated to get confirmation or can it be custom generated.

Analysis. TaskMaster is able to confirm specific details of the task before moving on to the next step. From the context, TaskMaster is able to generate dialogue responses to confirmation instructions on different aspects of the task to ensure the user has the correct information to complete a specific step. Conversely, Chefbot cannot confirm questions, there is either no understanding of the dialogue context and the sequence is forced to the next step, or in the case of question 7, the system is stuck in a ’Please repeat the question’ until a slot or intent is full-filled to move the conversation forward.

TaskMaster	ChefBot
Question. 7 **Recipe 23** U(1): **How many green onions do I need?** S(1): 1	Question. 7 **Recipe 23** U(1): **How many green onions do I need?** S(1): Please repeat the question
Question. 8 **Recipe 366** U(2): **What type of flour should I use?** S(2): all-purpose flour	Question. 8 **Recipe 366** U(2): **What type of flour should I use?** S(2): Now drain, crumble and set them aside, after that preheat the oven to 350 degrees Fahrenheit and lightly grease a 9 inch pie pan.
Question. 9 **Recipe 410** U(3): **How long do I boil the potato’s for?** S(3): 15 - 20 minutes	Question. 9 **Recipe 410** U(3): **How long do I boil the potato’s for?** S(3): Now, add 240ml of heavy cream, the roasted garlic, and a sprinkle of salt and black pepper to the small saucepan

Table 3: Example of Confirmations

4.5 Clarification Questions

Asking for clarification in traditional SDS is problematic due to the complexity of understanding underlying concepts, thus, repeating the same question is one of the most common methods of clarification [3]. In questions 10-12, clarification questions are the logic behind a specific action or additional information that may not be completely grounded in the original data. For instance, ”Why do I need to wear oven gloves?” or ”how do I hold this knife safely?”. These questions require context, for instance, because the pan is hot, which may or may not be fully grounded in previous dialogue and therefore require flexible dialogue management to answer, as shown in Table 4.

TaskMaster	ChefBot
Question. 10 **Recipe 10** U(1): **Why do I need to wear gloves when handling habanero peppers?** S(1): to easily protect the wearer’s hand from hot objects	Question. 10 **Recipe 10** U(1): **Why do I need to wear gloves when handling habanero peppers?** S(1): Please repeat the question
Question. 11 **Recipe 1** U(2): **Why do I need to flour the work surface?** S(2): breaks up any lumps	Question. 11 **Recipe 1** U(2): **Why do I need to flour the work surface?** S(2): Please repeat the question
Question. 12 **Recipe 19** U(3): **Do I need butter for the first steps of this recipe?** S(3): they do not call for butter	Question. 12 **Recipe 19** U(3): D**o I need butter for the first steps of this recipe?** S(3): Please repeat the question

Table 4: Example of Clarifications

Analysis. In this case study, we asked the system questions that may not be fully grounded in the previous dialogue, prompting the user to ask for clarification. For example, ’why do I need to do [objective A]’ or ’Why do I need [object] for [objective A]’. In Question 10, TaskMaster is able to get context from the underlying concepts in the dialogue to provide clarification. However, ChefBot is unable to understand the underlying context and generates 'Please repeat the question' until a slot or intent is full-filled to move the conversation forward. In Question 11 however, both SDS struggle to understand the context from the previous dialogue. Here, TaskMaster generated an incorrect response because the understanding of 'why you need to flour the surface to prevent the dough from sticking' was not mentioned in the dialogue for this recipe, or in other recipes or in the object knowledge dataset. Thus, the answer is compiled from the object knowledge dataset with reference to flour 'A flour sifter is a tool that breaks up any lumps'. Comparatively, ChefBot was unable to understand this question due to lack of context and generated a 'Please repeat the question' response. In Question 12, TaskMaster was able to clarify if an ingredient was required for a specific step in the recipe using context from the previous and proceeding dialogue. However, ChefBot was unable to differentiate between previously spoken dialogue and proceeding dialogue.

4.6 Unplanned Occurrences

Unexpected questions and unplanned instances (see Table 5) may occur during a real-world task and may change the conditions and prevent the user from proceeding, for example, 'the food is burning, what do I do?' or 'how do I prevent the dough from sticking to the table?'. In questions 13-15, external factors have prevented the user from continuing to the next step and advice must be given on how to correct unplanned occurrences.

TaskMaster	ChefBot
Question. 13 **Recipe 140** U(1): **The vegetables are really dirty what do I do?** S(1): wash	Question. 13 **Recipe 140** U(1): **The vegetables are really dirty what do I do?** S(1): Please repeat the question
Question. 14 **Recipe 250** U(2): **The sauce is sticking what do I do?** S(2): Continue to stir occasionally	Question. 14 **Recipe 250** U(2): **The sauce is sticking what do I do?** S(2): Please repeat the question
Question. 15 **Recipe 1** U(3): **The muffins are burning in the oven what do I do?** S(3): remove the muffins from the griddle and let them cool thoroughly	Question. 15 **Recipe 1** U(3): **The muffins are burning in the oven what do I do?** S(3): Please repeat the question

Table 5: Example of Unplanned Occurrences

Analysis. We explored how the systems handle unexpected instances that may not be referenced in the dialogue and may require further grounding from additional sources. In Question 13, an object is unusable in its current state, and the user needs to perform an action to use it in the task. From the context in the objects database, TaskMaster is able to understand the act of washing an object in order to clean it by cross-referencing items used for action; such as washing dirty ingredients in the object knowledge database. ChefBot was unable to answer the question as there was no recognisable intents or actions in the utterance. In Question 14, TaskMaster is able to answer a question on an unplanned event from previous dialogue in a different context, e.g. to stir in the ingredients. However, ChefBot is unable to understand this question as it is not sufficiently grounded in the data to accurately understand the logic of a specific action in a different context. In Question 15, we consider an unplanned scenario that needs immediate attention, e.g. burning food. TaskMaster is able to capture knowledge from the proceeding steps, [Remove the muffins from the griddle and let them cool thoroughly] and context from the object knowledge database on [oven]to generate a response. Chefbot was unable to understand the question due to the lack of context i.e. intents and actions needed to answer.

5 DISCUSSION

In this paper, we demonstrated a platform-independent SDS for robots, TaskMaster, that is able to overcome some of the limitations of rule-based dialogue management methods with canned text commonly used in HRI. We hope that TaskMaster will help promote the investigation of real-world applications for HRI as well as help answer research questions related to human-robot collaboration in real-world settings. Our findings show that TaskMaster shows flexibility in answering questions for clarifying and confirming different aspects of a task, relevant domain knowledge, alternative objects, referring to missing objects in dialogue and managing unplanned and unscripted instances. The results from our case studies support the potential application of TaskMaster in areas of HRI where accurate and flexible task-based spoken interaction is central to successfully completing real-world tasks. This is particularly significant in domains such as assistive robots for rehabilitation or caring for the elderly where a greater diverse range of real-world phenomena in communication may occur, such as, forgetting an instruction or responding to unplanned events. We addressed the issues of accessibility surrounding platform-dependant robotic systems and developed TaskMaster as a platform-independent, open-source system. To demonstrate deployability, we include a GitHub repository so roboticists can download, install and run TaskMaster locally.

In our analysis, we identified the limitations of TaskMaster in understanding the differences between intents with designated actions, such as intents with actions that move the conversation to another step and questions that reference an intent, such as [next], and utterances that include an intent, but are not answerable with the connect action, i.e. 'Do I finish basting the pie before I move onto the next step'. We aim to address this issue in future research by creating more flexible sequence mapping for intents. Finally, TaskMaster contributes to dialogue and response generation which we identified as an under-explored area of HRI. With this research, we moved beyond a simple template-based process when conversing and generating output, so roboticists can take advantage of the flexibility afforded by dialogue systems and NLG when generating output in dynamic environments or when output must be generated in different contexts or multiple target languages.

ACKNOWLEDGMENTS

This research is supported under the EPSRC projects NLG for low-resource domains (EP/T024917/1) and CiViL (EP/T014598/1). For

the purpose of open access, the author has applied a 'Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

REFERENCES

[1] 2022. *Natural Language in Human Robot Interaction (NLiHRI) at SIGDIAL 2022*. Association for Computational Linguistics, Edinburgh, Scotland. https://2022.sigdial.org/call-for-papers-nlihri/

[2] Eleni Adamopoulou and Lefteris Moussiades. 2020. Chatbots: History, technology, and applications. *Machine Learning with Applications* 2 (2020), 100006. https://doi.org/10.1016/j.mlwa.2020.100006

[3] Eleni Adamopoulou and Lefteris Moussiades. 2020. An Overview of Chatbot Technology. In *Artificial Intelligence Applications and Innovations*, Ilias Maglogiannis, Lazaros Iliadis, and Elias Pimenidis (Eds.). Springer International Publishing, Cham, 373–383.

[4] Samer Al Moubayed, Jonas Beskow, Gabriel Skantze, and Björn Granström. 2012. Furhat: A Back-Projected Human-Like Robot Head for Multiparty Human-Machine Interaction. In *Cognitive Behavioural Systems*, Anna Esposito, Antonietta M. Esposito, Alessandro Vinciarelli, Rüdiger Hoffmann, and Vincent C. Müller (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 114–130.

[5] Nivedita Balu and Jeffrey Dastin. 2021. Amazon launches robot to roll around house, Disney resort voice assistant. (2021).

[6] Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open Source Language Understanding and Dialogue Management. https://doi.org/10.48550/ARXIV.1712.05181

[7] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the Opportunities and Risks of Foundation Models. https://doi.org/10.48550/ARXIV.2108.07258

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. https://doi.org/10.48550/ARXIV.2005.14165

[9] Brian Davis, Yvette Graham, John Kelleher, and Yaji Sripada (Eds.). 2020. *Proceedings of the 13th International Conference on Natural Language Generation*. Association for Computational Linguistics, Dublin, Ireland. https://aclanthology.org/2020.inlg-1.0

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/ARXIV.1810.04805

[11] Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2022. A Survey of Natural Language Generation. *ACM Comput. Surv.* (jul 2022). https://doi.org/10.1145/3554727 Just Accepted.

[12] Marlena R. Fraune, Iolanda Leite, Nihan Karatas, Aida Amirova, Amélie Legeleux, Anara Sandygulova, Anouk Neerincx, Gaurav Dilip Tikas, Hatice Gunes, Mayumi Mohan, Nida Itrat Abbasi, Sudhir Shenoy, Brian Scassellati, Ewart J. de Visser, and Takanori Komatsu. 2022. Lessons Learned About Designing and Conducting Studies From HRI Experts. *Frontiers in Robotics and AI* 8 (2022). https://doi.org/10.3389/frobt.2021.772141

[13] Nadine Glas and Catherine Pelachaud. 2015. Definitions of engagement in human-agent interaction. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*. 944–949. https://doi.org/10.1109/ACII.2015.7344688

[14] 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. https://doi.org/10.48550/ARXIV.2209.06321

[15] Lucrezia Grassi, Carmine Tommaso Recchiuto, and Antonio Sgorbissa. 2022. Sustainable Verbal and Non-verbal Human-Robot Interaction Through Cloud Services. https://doi.org/10.48550/ARXIV.2203.02606

[16] Bahar Irfan, Mehdi Hellou, and Tony Belpaeme. 2021. Coffee With a Hint of Data: Towards Using Data-Driven Approaches in Personalised Long-Term Interactions. *Frontiers in Robotics and AI* 8 (09 2021), 676814. https://doi.org/10.3389/frobt.2021.676814

[17] Kristiina Jokinen and Graham Wilcock. 2012. Multimodal Open-Domain Conversations with the Nao Robot. In *Natural Interaction with Robots, Knowbots and Smartphones, Putting Spoken Dialog Systems into Practice.*

[18] Jurgita Kapočiūtė-Dzikienė. 2020. A Domain-Specific Generative Chatbot Trained from Little Data. *Applied Sciences* 10 (03 2020), 2221. https://doi.org/10.3390/app10072221

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. https://doi.org/10.48550/ARXIV.1907.11692

[20] Davide Mulfari, Antonio Celesti, Maria Fazio, Massimo Villari, and Antonio Puliafito. 2014. Using Embedded Systems to Spread Assistive Technology on Multiple Devices in Smart Environments. *Proceedings - 2014 IEEE International Conference on Bioinformatics and Biomedicine, IEEE BIBM 2014*. https://doi.org/10.1109/BIBM.2014.6999234

[21] Amal Nanavati, Nick Walker, Lee Taber, Christoforos Mavrogiannis, Leila Takayama, Maya Cakmak, and Siddhartha Srinivasa. 2022. Not All Who Wander Are Lost: A Localization-Free System for In-the-Wild Mobile Robot Deployments. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction* (Sapporo, Hokkaido, Japan) *(HRI '22)*. IEEE Press, 422–431.

[22] University of Glasgow. 2022. GRILLBot: A flexible conversational agent for solving complex real-world tasks. In *Alexa Prize TaskBot Challenge Proceedings*. https://www.amazon.science/alexa-prize/proceedings/grillbot-a-flexible-conversational-agent-for-solving-complex-real-world-tasks

[23] Amit Kumar Pandey and Rodolphe Gelin. 2018. A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of Its Kind. *IEEE Robotics Automation Magazine* 25, 3 (2018), 40–48. https://doi.org/10.1109/MRA.2018.2833157

[24] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. https://doi.org/10.48550/ARXIV.1910.10683

[25] Laurel D. Riek. 2012. Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. *J. Hum.-Robot Interact.* 1, 1 (jul 2012), 119–136. https://doi.org/10.5898/JHRI.1.1.Riek

[26] Finn Rietz, Alexander Sutherland, Suna Bensch, Stefan Wermter, and Thomas Hellström. 2021. WoZ4U: An Open-Source Wizard-of-Oz Interface for Easy, Efficient and Robust HRI Experiments. *Frontiers in Robotics and AI* 8 (2021). https://doi.org/10.3389/frobt.2021.668057

[27] Jaime A. Rincon, Angelo Costa, Paulo Novais, Vicente Julian, and Carlos Carrascosa. 2019. A New Emotional Robot Assistant That Facilitates Human Interaction and Persuasion. *Knowl. Inf. Syst.* 60, 1 (jul 2019), 363–383. https://doi.org/10.1007/s10115-018-1231-9

[28] Bayu Setiaji and Ferry Wibowo. 2016. Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling. 72–77. https://doi.org/10.1109/ISMS.2016.53

[29] Syamimi Shamsuddin, Luthffi Idzhar Ismail, Hanafiah Yussof, Nur Ismarrubie Zahari, Saiful Bahari, Hafizan Hashim, and Ahmed Jaffar. 2011. Humanoid robot NAO: Review of control and motion exploration. In *2011 IEEE International Conference on Control System, Computing and Engineering*. 511–516. https://doi.org/10.1109/ICCSCE.2011.6190579

[30] Sichao Song, Jun Baba, Junya Nakanishi, Yuichiro Yoshikawa, and Hiroshi Ishiguro. 2022. Costume vs. Wizard of Oz vs. Telepresence: How Social Presence Forms of Tele-Operated Robots Influence Customer Behavior. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction* (Sapporo, Hokkaido, Japan) *(HRI '22)*. IEEE Press, 521–529.

[31] Micol Spitale, Chris Birmingham, R. Michael Swan, and Maja J Matarić. 2021. Composing HARMONI: An Open-source Tool for Human and Robot Modular OpeN Interaction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 3322–3329. https://doi.org/10.1109/ICRA48506.2021.9560992

[32] Carl Strathearn. 2021. Artificial Eyes with Emotion and Light Responsive Pupils for Realistic Humanoid Robots. *Informatics* 8, 4 (2021). https://doi.org/10.3390/informatics8040064

[33] Carl Strathearn and Dimitra Gkatzia. 2020. The Task2Dial Dataset: A Novel Dataset for Commonsense-enhanced Task-based Dialogue Grounded in Documents. In *Proceedings of The Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021)*. Association for Computational Linguistics (ACL), 242–251. http://researchrepository.napier.ac.uk/Output/2807732 "Note that all accepted papers will be published in ACL Anthology. In addition, selected papers will be published in Signals and Communication Technology (Springer), which is indexed in Scopus." - from http://icnlsp.org/icnlsp2021/.

[34] Carl Strathearn and Dimitra Gkatzia. 2021. Chefbot: A Novel Framework for the Generation of Commonsense-enhanced Responses for Task-based Dialogue Systems. In *Proceedings of the 14th International Conference on Natural Language Generation*. Association for Computational Linguistics, Aberdeen, Scotland, UK, 46–47. https://aclanthology.org/2021.inlg-1.5

[35] Ma E.M. Strathearn, C. 2021. A Novel Speech to Mouth Articulation System for Realistic Humanoid Robots. In *J Intell Robot Syst 101, 54 (2021)*. https://doi.org/10.1007/s10846-021-01332-2

[36] Carnegie Mellon University. 2022. Tartan: A taskbot that assists with recipes and do-it-yourself projects. In *Alexa Prize TaskBot Challenge Proceedings*. https://www.amazon.science/alexa-prize/proceedings/tartan-a-taskbot-that-assists-with-recipes-and-do-it-yourself-projects

[37] National Taiwan University. 2022. Miutsu: NTU's TaskBot for the Alexa Prize. In *Alexa Prize TaskBot Challenge Proceedings*. https://www.amazon.science/alexa-prize/proceedings/miutsu-ntus-taskbot-for-the-alexa-prize

[38] Texas AM University. 2022. Howdy Y'all: An Alexa TaskBot. In *Alexa Prize TaskBot Challenge Proceedings*. https://www.amazon.science/alexa-prize/proceedings/howdy-yall-an-alexa-taskbot

[39] Graham Wilcock and Kristiina Jokinen. 2021. *Towards Increasing Naturalness and Flexibility in Human-Robot Dialogue Systems*. Springer Singapore, Singapore, 109–114. https://doi.org/10.1007/978-981-15-9323-9_9

[40] Baijun Xie and Chung Hyuk Park. 2021. Empathetic Robot With Transformer-Based Dialogue Agent. In *2021 18th International Conference on Ubiquitous Robots (UR)*. 290–295. https://doi.org/10.1109/UR52253.2021.9494669

[41] Fengpei Yuan, Amir Sadovnik, Ran Zhang, Devin Casenhiser, Eun Jin Paek, and Xiaopeng Zhao. 2022. A simulated experiment to explore robotic dialogue strategies for people with dementia. *Journal of Rehabilitation and Assistive Technologies Engineering* 9 (2022), 20556683221105768. https://doi.org/10.1177/20556683221105768 arXiv:https://doi.org/10.1177/20556683221105768 PMID: 35692231.

[42] Hainan Zhang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2018. Tailored Sequence to Sequence Models to Different Conversation Scenarios. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 1479–1488. https://doi.org/10.18653/v1/P18-1137