HIGH SCHOOL EDITION
Journal of Student Research
Volume 14 Issue 1 (2025)

On the Potential of Using Artificial Intelligence to Handle Healthcare Instructions for the Elderly

Yamaan Khundakjie[1] and Gregory Theos[#]

[1]Advanced Math and Science Academy Charter School, USA
[#]Advisor

ABSTRACT

Telehealth services have been playing an increasing role in the communication and management of healthcare. Cognitive decline in the elderly presents challenges for the effectiveness of these services. This research shares findings from the development of a smartphone app prototype that can reduce select burdens on the growing population of the aging-in-place elderly, with normal-aging cognition or with mild cognitive impairment (MCI), when receiving healthcare instructions. These burdens include comprehending and summarizing instructions, extracting actions, and creating planning reminders. The app accepts audio input representing instructions spoken by healthcare professionals. After performing speech-to-text transcription, the app applies artificial intelligence (AI) classification, natural language processing and summarization capabilities to automatically produce action reminders. Audio samples varying in complexity (number and order of actions) and speech speed are used to test the app. This categorization is inspired by the established distinct deterioration of executive functions (e.g. planning, working memory, and sustained attention) and temporal information processing that contribute to cognitive aging. The app's processing speed as well as its accuracy and completeness of automatic creation of action reminders are measured. When compared to several widely available classic and AI-powered assistants of similar or partial functionality in smartphones, the results demonstrated notable deficiencies in existing assistants. This includes a decreasing probability of capturing three or more actions and an increased probability of replays for larger audio samples. Conclusions from this research call for an increased attention in the commercial and scientific domains to combat cognitive challenges affecting the elderly's handling of healthcare instructions.

Introduction

Adults, including the elderly, are spending more time at home. This behavior is a long-term trend (Sharkey, 2024) that has been further amplified by the *corona virus disease of 2019* (COVID-19). At the same time, common usage of smartphones by the elderly has increased to levels that close the gap with those thirty and younger (Faverio, 2022). Such increased usage can span many activities unrelated to healthcare and was shown to have attenuated levels of adverse psychological impact when compared to those studied in younger people (Busch, Hausvik, Ropstad, & Pettersen, 2021).

The increase of smartphone usage by the elderly should not imply that the current technology has always been built and optimized with this growing segment of the population in mind as it has been for the younger population (Iancu & Iancu, 2020). One important area of smartphone or smart-device utilization by aging adults is *telehealth*. This area refers to the delivery and management of remote healthcare services via electronic and telecommunications means. Early and prevalent attitudes remain positive about the potential of telehealth benefits to healthcare professionals, infrastructure, and patients (Vockley, 2015) especially in the case of chronic conditions (P, et al., 2018). However, careful attention to the common barriers affecting successful utilization of telehealth by the elderly is warranted (C, et al., 2020). A select subset of those barriers are enforced by cognitive decline such as mental acuity and technology complexity. This research effort attempts to focus on handling the healthcare instructions received in audio

form (e.g. in voicemail or live conversations) by aging-in-place adults with cognitive decline to reduce their cognitive burden and improve productivity in telehealth.

The objective of this research is two-fold. Firstly, beneficial capabilities of artificial intelligence (AI) to offload select cognitive burdens on the elderly are investigated. The target elderly population is those aging in place without a caregiver and considered to be at normal-aging cognition levels or *mild cognitive impairment* (MCI) (Petersen & Negash, 2008), characterized by deteriorations below dementia. This investigation is in the form of prototyping an Android® app, named *Medical Understanding Nexus Assistant* (MUNA), to handle the use case of comprehending and acting on commonly received healthcare instructions. Secondly, a comparative assessment of the app's performance is conducted against common as well as recently introduced AI-powered personal smartphone assistants. This assessment sheds light on the level of attention needed to enable the elderly to compensate for cognitive decline while using telehealth.

The approach taken in this research focuses on a smart-device requiring only Internet access and a cloud-based conversational AI engine such as ChatGPT®. The device's own speech-to-text technology is used locally to transcribe audio. ChatGPT® (3.5 Turbo and 4o) is used for action identification and sample summarization. Finally, the device's local calendar app is used to create action reminders. This approach prioritizes availability, autonomy, and affordability for the elderly. Hence, the app prototype does not expect any caregiver intervention nor does it require the integration with proprietary or public healthcare infrastructure systems.

While inspired by generalized frameworks used to assess cognitive abilities such as the *Tower of London* (TOL) and its derivatives (Shallice, 1982), this research does not deploy them. Instead, specific experiments are purposely defined to explore the problem of handling healthcare instructions around two dimensions. The first dimension relates to *executive functions* which are generally the cognitive processes that guide goal-oriented behavior including planning, sequencing, working memory, and decision making. This dimension is investigated using healthcare audio samples of varying complexity represented by the number, schedule, and inter-relationship of actions requested of the elderly. The second dimension relates to temporal information processing speed (K, et al., 2016) and is investigated through varying speeds of speech of the audio samples. Consider for example the following sample:

"*Hi John! This is Elaine from the Gastroenterology Center at UMASS Health. Just a quick reminder about your endoscopy appointment tomorrow at 3:30 PM. Please remember to stop eating 8 hours beforehand and stop drinking 6 hours before the appointment. If anything's unclear or you have any questions, give me a call at 617-345-1256.*"

In the above sample, there are 3 requested actions that have a schedule relationship. The questions investigated in this research are:

- Can AI be used to identify 3 actions and create 3 calendar reminders for the elderly automatically?
- Are the reminders accurate based on the original audio sample?
- How is the accuracy and number of reminders affected if the audio is heard at different speeds?
- How well will samples of lesser or higher complexity be handled?

Related Work

At the time of this research several searches for apps in the Android® and iOS® app stores did not show any active apps capable of the end-to-end functionality of processing a healthcare audio sample to automatically generate calendar reminders in the device of the elderly.

Several apps are capable of summarizing audio samples such as (OtterAI, 2024), (SoundTypeAI, 2024), and (Summerizer, 2024) using AI. Other healthcare-oriented apps focus on manual textual or voice entry by the user for each action related to calendar reminders, prescription dispensing, and appointment scheduling such as (SetTimeAppointmentScheduler, 2024) which uses voice entry for medication and (SetmoreAppointmentScheduling, 2024).

In addition, several healthcare apps are available to improve the productivity of healthcare professionals and manage their appointments. Such apps are not concerned with managing reminders in one place for the aging patient's interactions with different healthcare professionals and facilities.

Common personal assistants have the basic capabilities to be instructed by users through voice to perform actions on their smart devices such as Google Assistant®, Amazon Alexa®, and Apple Siri®. However, such assistants are focused on completing a single instruction or instructions related to smart-home device controls and lack conversational AI capabilities. More recent advanced versions such as Google Gemini® and Apple Siri 2.0® with Apple Intelligence® introduce conversational AI capabilities with the ability to perform more than one action. In this research, we assess the efficacy of existing personal assistants using the same set of experiments by which we evaluate the app prototype MUNA.

Methods and Materials

This section discusses the process of designing and developing the app prototype followed by the design and execution of the experiments to measure its efficacy.

Design Considerations

The target of this app is the elderly population aging in place with aging cognition decline at or below MCI. More advanced levels of cognitive decline associated with dementia or Alzheimer's and those requiring frequent caregiver intervention present a unique set of requirements and use cases warranting different design approaches in telehealth technology. In addition, to initially focus this research on cognitive decline, auditory and visual impairments are not considered in the current app prototype.

Accommodating privacy regulations is not considered in the prototype. The investigation objectives are met without actively recording any humans conversing through a phone call or in person. Instead, prepared audio samples synthesized from realistic healthcare instruction messages are played to all apps.

While iPhone® is considered to be the most popular smartphone in the United States, for the purpose of answering the main research questions, the prototype is developed in Kotlin® for Android®. The are two reasons for this choice. Firstly, the Android Studio® platform used for development provides an open-source experimentation environment that is more widely accessible for developers on multiple types of devices. Secondly, the conversational AI capabilities possible in Android® devices have been known to be superior to those in iPhone® devices for many years.

Prototype Design and Development Platform

The high-level data flow for the app prototype MUNA is shown in Figure 1. Android® built-in speech-to-text library is used to transcribe the audio sample representing healthcare instructions. The number of instructions ranges from having 1-5 actions and is never 0. If the MUNA is unable to identify any actions, the audio sample text is only summarized by ChatGPT® and stored locally with no reminders created. If the instructions contain actions, the actions are sent to another instance of ChatGPT® which parses them into a form that can be taken in by the Calendar app. Then, reminders are created for each action, and a summary is also stored locally.

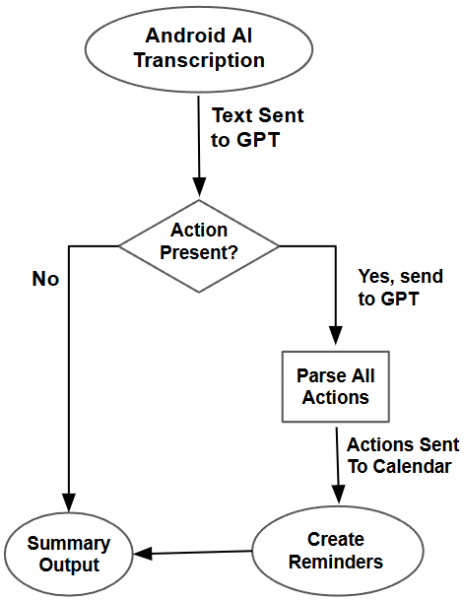

Figure 1. Overall, high-level flowchart of app functionality

The classification of whether a text contains any actions makes use of ChatGPT-3.5 Turbo®. After MUNA receives a transcript from Android® speech-to-text it sends it to ChatGPT-3.5 Turbo® with a specific pre-prompt shown in Figure 2. This is done to optimize the overall response time by using this lighter weight GPT instance instead of the more powerful GPT-4o® that consumes more time and resources for a step that does not involve intense processing or structuring of data.

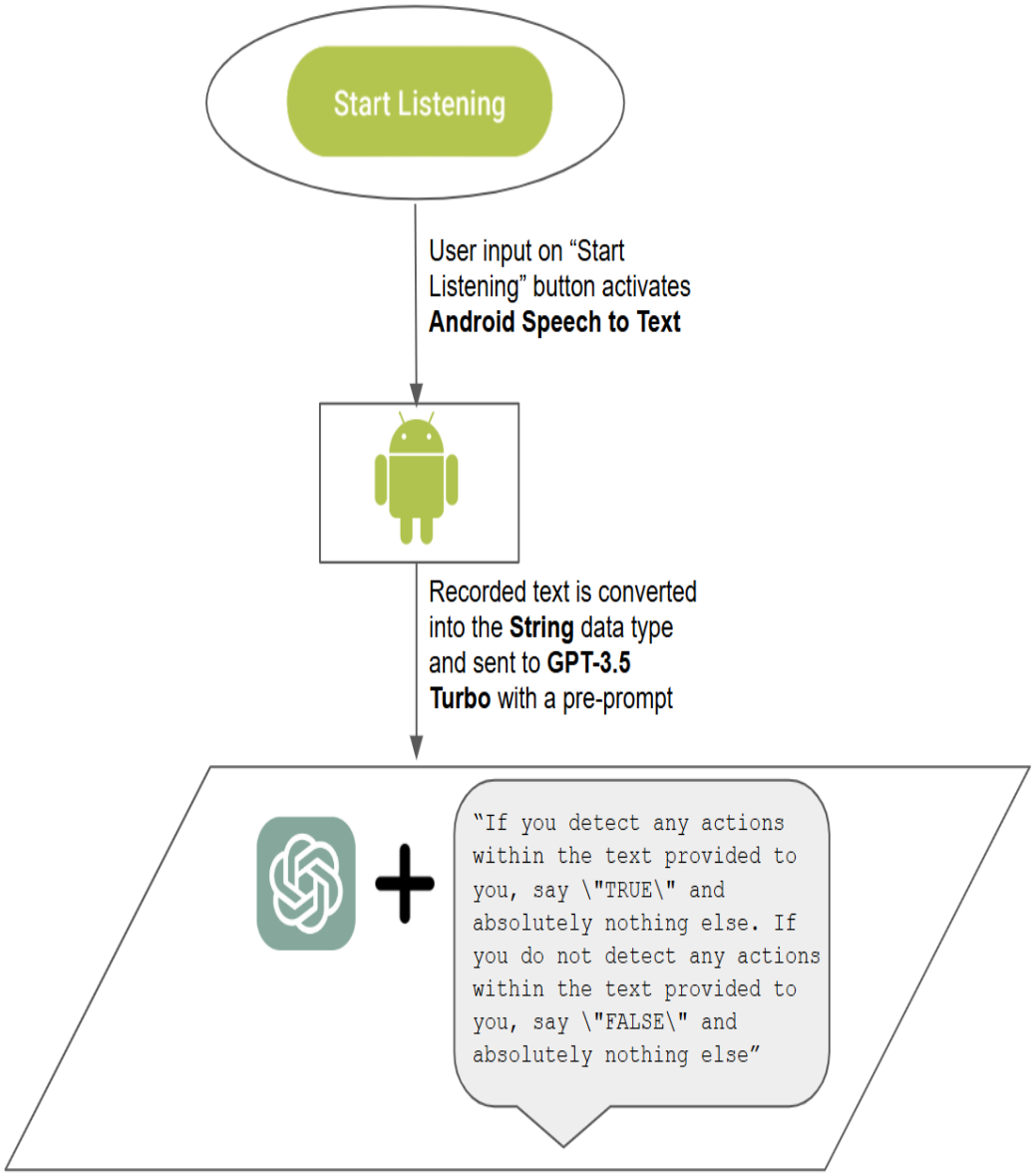

Figure 2. Detailed flowchart of transcription to action detection Steps

This result from this first step is either "TRUE" or "FALSE" due to the tuned pre-prompt. "TRUE" indicates actions exist within the text, and "FALSE" indicates there are none. If actions are detected, MUNA moves to the next layer of AI-processing, where more difficult and precise tasks are asked of the ChatGPT® API. Specifically, the text is sent to GPT with a tuned pre-prompt instructing it to parse each element of the action (e.g. date, time, title, etc.) into a specifically structured list that can later be sent to the Calendar app for reminder creation.

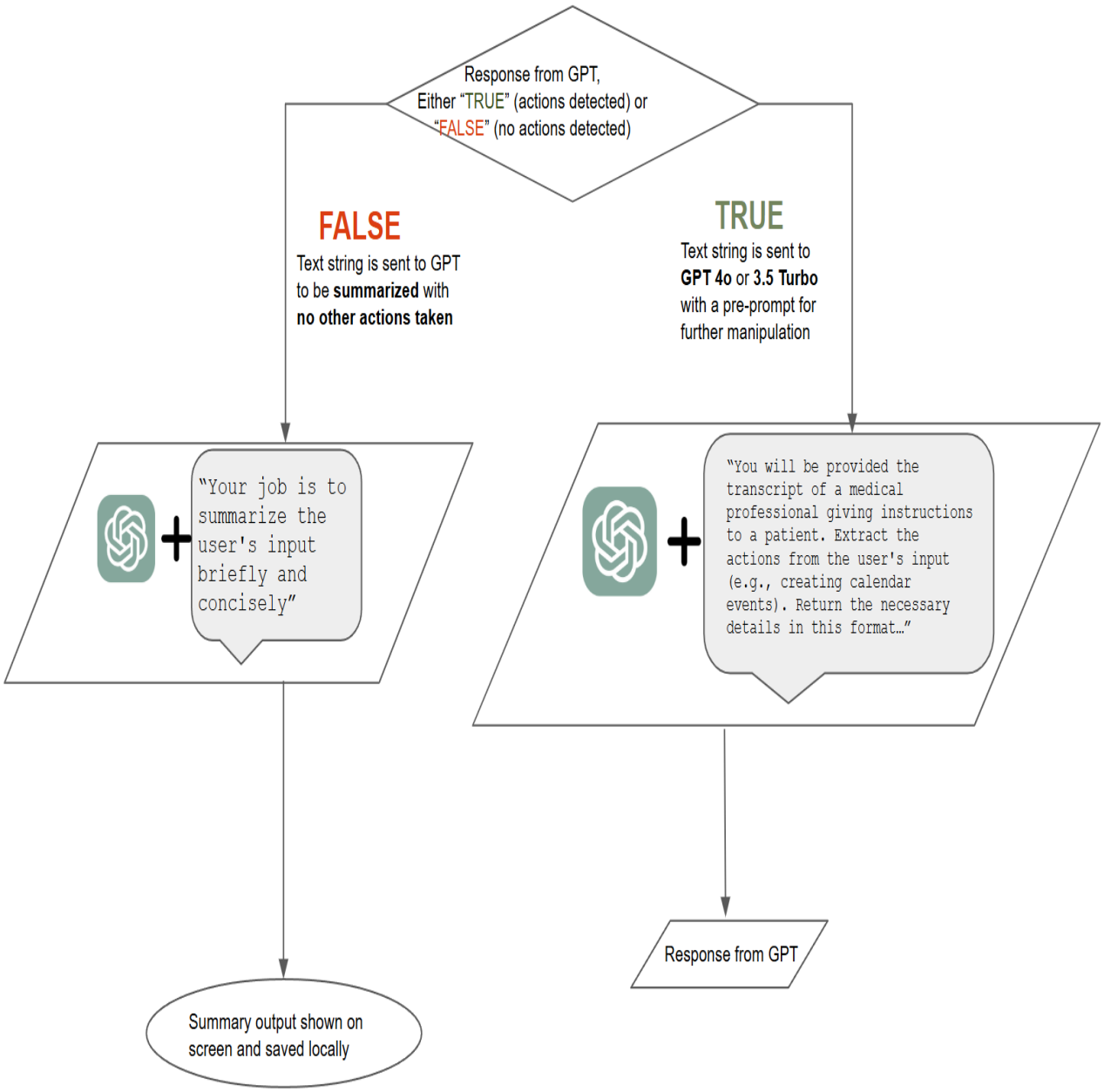

Figure 3. Detailed flowchart of GPT summary creation and/or reminder parsing preparation

As shown in Figure 3, this is the step in which both models of ChatGPT® are investigated to assess their efficacy in action extraction and reminder structured text formatting. Once GPT completes action parsing and outputs a list of reminder attributes, MUNA interacts with the Calendar app to create reminders using this list.

ChatGPT® communication with MUNA was facilitated with an affordable ChatGPT API subscription. The message to ChatGPT® included a guiding prompt that needed several iterations to produce the desired outcome. The summarization task is done by GPT after the reminders are created to avoid increasing the action reminder creation time observed by the user irrespective of "TRUE" vs "FALSE" case as displayed in Figure 4.

The development platform used for this app was Android Studio Koala | 2024.1.1 Patch running under Windows 10® on a personal computer equipped with Intel® Core i5-14600K and 32GB of DDR5 memory.

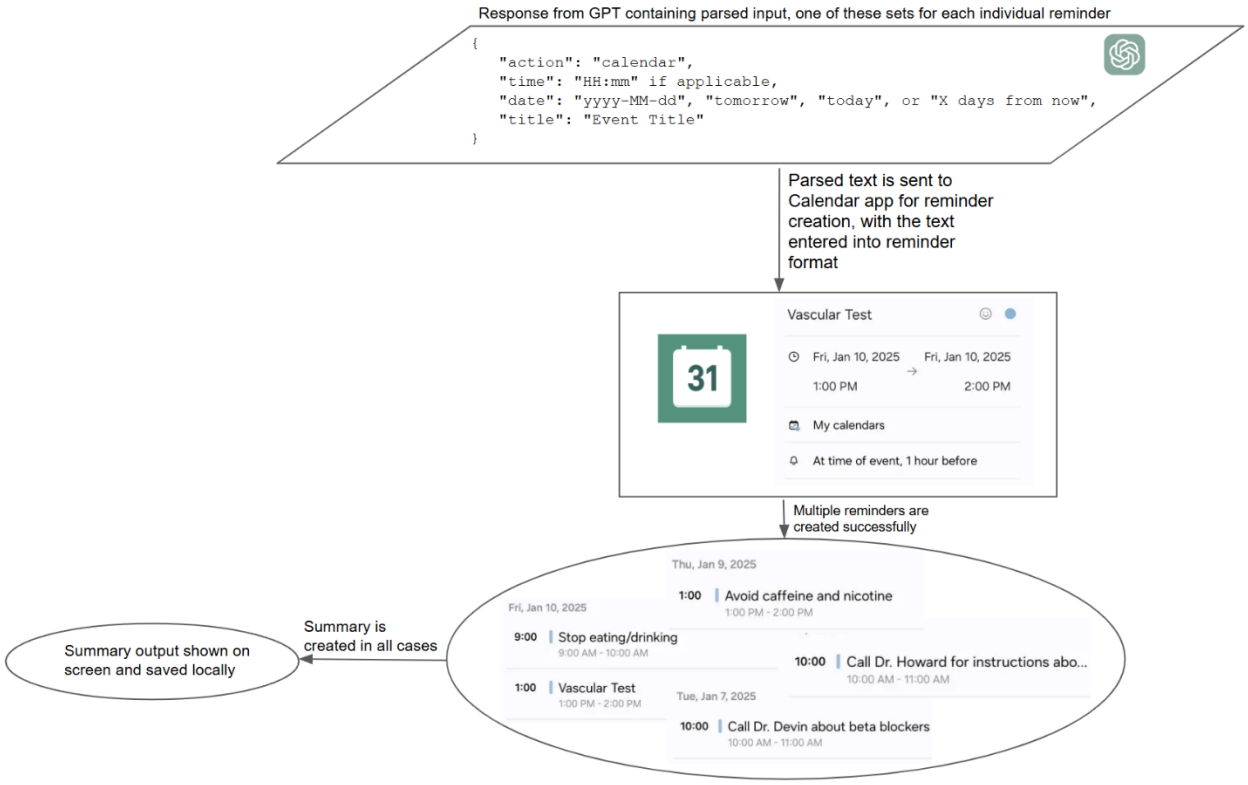

Figure 4. Flowchart of Final App Steps Involving Calendar Communication and Summarization

Experiment Design and Procedures

To measure the efficacy of MUNA, several experiments were designed around processing audio samples of 4 levels of complexity spoken at 3 different speeds. As mentioned earlier, this design intends to explore the elements of cognitive decline concerning executive functions which relate to the complexity levels as well as temporal information processing which relates to speech speed and task sequencing. The text for the different samples is shown in Table 1.

Table 1. Text for audio samples representing 4 levels of complexity

Sample 1 (Complexity L1)	Hi Sarah! This is Joel from Dr. Dean's office at Aim Family Dental. I just wanted to remind you to take 3000mg of antibiotics at 2:00 PM tomorrow, before your dental implant surgery. If you have any questions, feel free to give me a call at 508-932-2814.
Sample 2 (Complexity L2)	Hi John! This is Elaine from the Gastroenterology Center at UMASS Health. Just a quick reminder about your endoscopy appointment tomorrow at 3:30 PM. Please remember to stop eating 8 hours beforehand and stop drinking 6 hours before the appointment. If anything's unclear or you have any questions, give me a call at 617-345-1256.
Sample 3 (Complexity L3)	Hello Lisa! This is Jesse calling from Doctor Paige's office at the Bone Densitometry Center at Lawrence General Hospital. This is a reminder for your bone density exam scheduled for 11 AM on February 13th, 2025. Starting 24 hours before your exam,

	please make sure to not take any calcium supplements. On the exam day wear comfortable clothes without zippers or metal. If you need any more information, feel free to reach out to me at 508-777-7451.
Sample 4 (Complexity L4)	Hi Bill! This is Katherine from Doctor King's office in the Heart Care department at UMASS Memorial. I'm calling with a few instructions for your non-invasive vascular test on January 10th, 2025, at 1 PM. Try to avoid caffeine and nicotine for 24 hours before the test, and don't eat or drink for at least 4 hours before. Since you're on diabetes medication, please contact Doctor Howard for specific guidance about two weeks before. Regarding your beta blockers and heart medications, give Doctor Devin a call three days before the test to confirm if you should pause them. If you have any questions at all, just call me at 978-128-1238.

The first complexity level requires only a single action from the patient. The second level includes 3 actions at explicit times phrased in relative relationship to each other. The third level requires 4 actions to be taken farther out in the future from the time of receiving the message. The fourth level requires 5 actions to be taken with each challenge described for the previous levels also being present. Healthcare messages longer than level 4 are assumed to be broken into smaller messages or are impractical for audio delivery for most patients regardless of age and likely to be delivered in textual form.

Each of the samples was audio-synthesized as three speech speeds: slow, medium, and fast. The slow speech speed was set at 0.7x the medium speech speed with the fast speech speed being 1.3x the medium speech speed. These levels were a careful trade-off between realistic speech patterns and technology limitations such as speech-to-text technology timeout behaviors on very slow speeds. Audio synthesis was done using the technology offered by https://speechgen.io/ with following settings:

- default pitch
- pause for paragraphs = NA
- pause for sentences = 150 ms
- default sample rate

The following assistants were assessed in addition to MUNA using the same audio samples: Siri® from Apple®, Google Assistant® from Google®, Gemini® from Google®, and Alexa® from Amazon®. The last two are considered the latest available conversational personal assistants while the first two did not have conversational capabilities. At the time of this investigation, Apple Intelligence® and Siri 2.0® were not broadly available and could not be included in the experiments. As noted under section Experimental Limitations the small number of existing assistants that target this research's problem area and lack of randomness in their responses to audio samples reduce this research's ability to present detailed statistical analyses including confidence intervals and tests of significance. The statistical analysis presented later does not assume a normal distribution.

The experimentation platforms consisted of two platforms. All compared assistants except Siri® were tested on a Samsung Galaxy S24® running Android 14® while Siri® was tested on Apple iPhone 15 ProMax® running iOS 18.1®. Each sample is played from a Lenovo Yoga 7 14IRL8 laptop.

Experiment Terminology and Measures

The following terms are used when discussing the experiment's inputs

Sample: An audio message containing a set of healthcare instructions.

Prompting: The act of playing the sample into the microphone of each device that is running an assistant.

Pre-prompt: A set of instructions that will be provided to each assistant before it is prompted. These instructions are used to explain the context of the sample it will receive and what output it must produce. For example, a pre-prompt can be along the lines of *"In this next message, I will provide a set of healthcare instructions that were given*

to me. Your job is to extract actions I must take from these instructions and create one or multiple reminders for each action ". These instructions will not be counted in "Average Number of Prompts Before any Success" (explained next). Different assistants may require different pre-prompts to process a prompt to maximize their abilities due to their unique designs.

The following measurements are taken for each audio sample:

Action Reminder Creation Average Time (ARCAT): The time it takes an assistant in seconds to complete a speech-to-text transcription and create action reminders in the calendar. In other words, this is the difference between the time the assistant registers the instructions after they are done being played and the reminder creation time. The resolution is one tenth of a second.

Average Number of Missed Actions (ANMA): The average number of actions an assistant completely excluded from any reminders created. This does not consider the number of reminders created. If an action name is present in any reminder, it is not considered missed.

Average Number of Errors in Reminders (ANEIR): The average number of times an action's attribute is captured incorrectly, such as the date or time of a reminder being incorrect. This also considers the case when the names of different actions are bundled into one reminder title. If the actions in this case are meant to have separate times yet are all placed under one time, then errors are counted for each action not matching the time of the reminder.

Average Number of Prompts Before any Success (ANPBS): The average number of prompts needed before the first reminder is created, irrespective of the number of reminders created at once. The pre-prompt that comes before each set of instructions is excluded from this measure, as all assistants are measured while having context of the task they must complete. This measure includes, but is not limited to, replaying the instructions if no reminder is created or if the assistant requests it, replaying a certain part of the instructions for information the assistant missed, or replaying the instructions if the assistant's speech-to-text listening mechanism cuts off for any reason.

Experiment Procedures

Measurements were collected from playing a total of 12 audio samples (4 levels x 3 speeds) to each app at maximum audio level. The smartphone running the assistant under test is placed next to the laptop's speakers with no gaps. The environment where measurements are collected is a study room without ambient noise.

For ARCAT, reminder creation time was measured from the point the audio sample is received until any action reminders are created as indicated by a message from the assistant or a notification from the Calendar app. ANMA was measured by visually inspecting the reminders created in the calendar app and comparing against expected actions requested by the sample's instructions. ANEIR was measured by visually inspecting created reminders for compliance with action attributes present in each sample. Finally, ANPBS was measured by manually counting the prompts needed before any success in creating any reminders.

After each audio sample experiment, the calendar app was cleared of all reminders and the assistant under test was closed.

Since MUNA was designed specifically to compensate for cognitive decline, no pre-prompting was performed.

Results

This section summarizes ARCAT, ANMA, ANEIR, and ANPBS for each assistant individually followed by a comparative set of results. Four charts are shown for individual results each presenting results across the levels of complexity L1-L4 and speech speed. Each measure was tested with n = 3 trials (see Experimental Limitations), where the average for each measurement was taken across all trials. In cases where a measurement was not possible, the result is indicated as #N/A. For example, if an assistant timed out in action creation at a certain complexity level and speed, then #N/A is presented as the result for average number of missed action reminders.

Apple Siri® Performance and Observations

As shown in Figure 5, ARCAT for Siri® ranges from 1.1 sec. to 1.6 sec, with no significant correlation to the characteristics of the audio samples. However, it appears that its reaction time is confined to the narrow range above for all samples. To understand ANMA and ANEIR results, it is important to note that Siri® was observed to create a single action reminder from the entire transcription irrespective of the actual number of actions requested in the sample.

At L1 complexity, the results were ideal. However, at L2, even though misses were graded at zero due to creating a reminder that includes all actions, the placement of two out of three L2 actions in the same reminder was counted as two mistakes.

At L3, Siri® encountered high action reminder misses because when it accepted a prompt, the reminders were created with content that cut-off the rest of the sample's text possibly due to limitations on the length of a reminder title text in iOS. In addition, Siri® also often responded with "for who?" even after providing it explicit instructions to create reminders from the healthcare instructions, not send them to anyone. Some tests had to be repeated due to this behavior. For male voice samples, it was possible to move past this behavior. Siri® performed the worst at L4. It was not able to create a single reminder or understand instructions. Overall, it did better with shorter and more concise tasks.

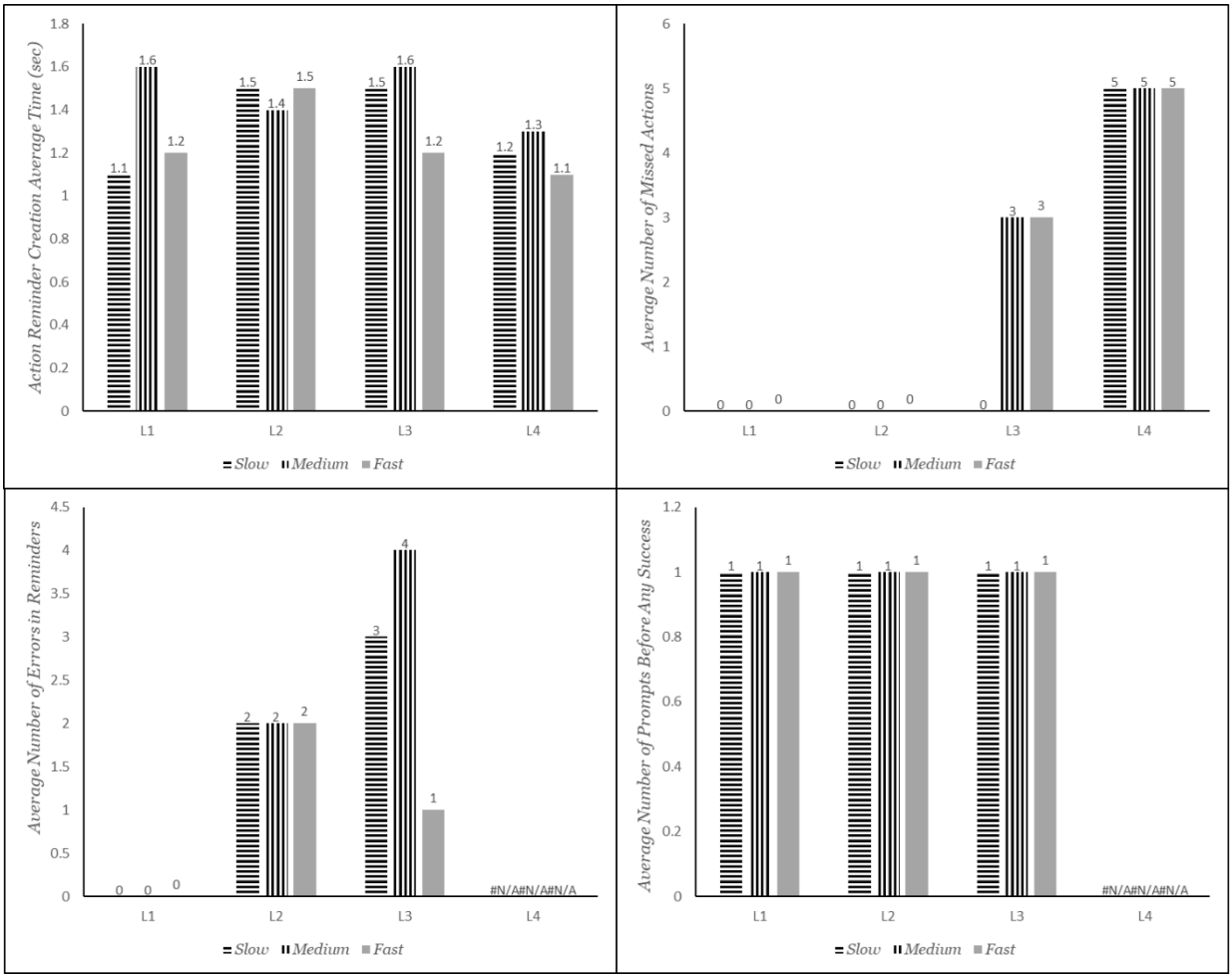

Figure 5. Siri® app performance measurements for each audio sample (L1-L4) and each sample speech speed

Google Assistant® Performance and Observations

Google Assistant® results are shown in Figure 6. ARCAT results are volatile and do not present any trends across samples. It was also observed that regardless of the sample, this assistant was only capable of creating a single action reminder. It frequently timed-out but when it did work, it usually required several prompts. For one case, it even required four prompts just for a reminder to be created.

For ANMA results, Google Assistant® followed a consistent trend of missing more actions as difficulty increased. Many of these misses were caused by its failure to create a reminder no matter how many prompts it was provided.

ANEIR results followed an increasing trend as sample difficulty increased. At L4, no data was collected for slow and medium speeds, as Google Assistant® failed to create any reminders. At fast speed, however, a reminder was created in only one of the trials, but one out of five actions requested at L4 were picked up. There were no errors present in this single reminder with the action.

ANPBS followed an unusual trend of decreasing as sample difficulty increased. This means that Google Assistant® required less prompts to create more complex reminders. It is possible that Google Assistant® may have a certain resource consumption barrier for processing more complex instructions. This could cause it to fulfill requests from the user with less clarifying questions or need for more prompts.

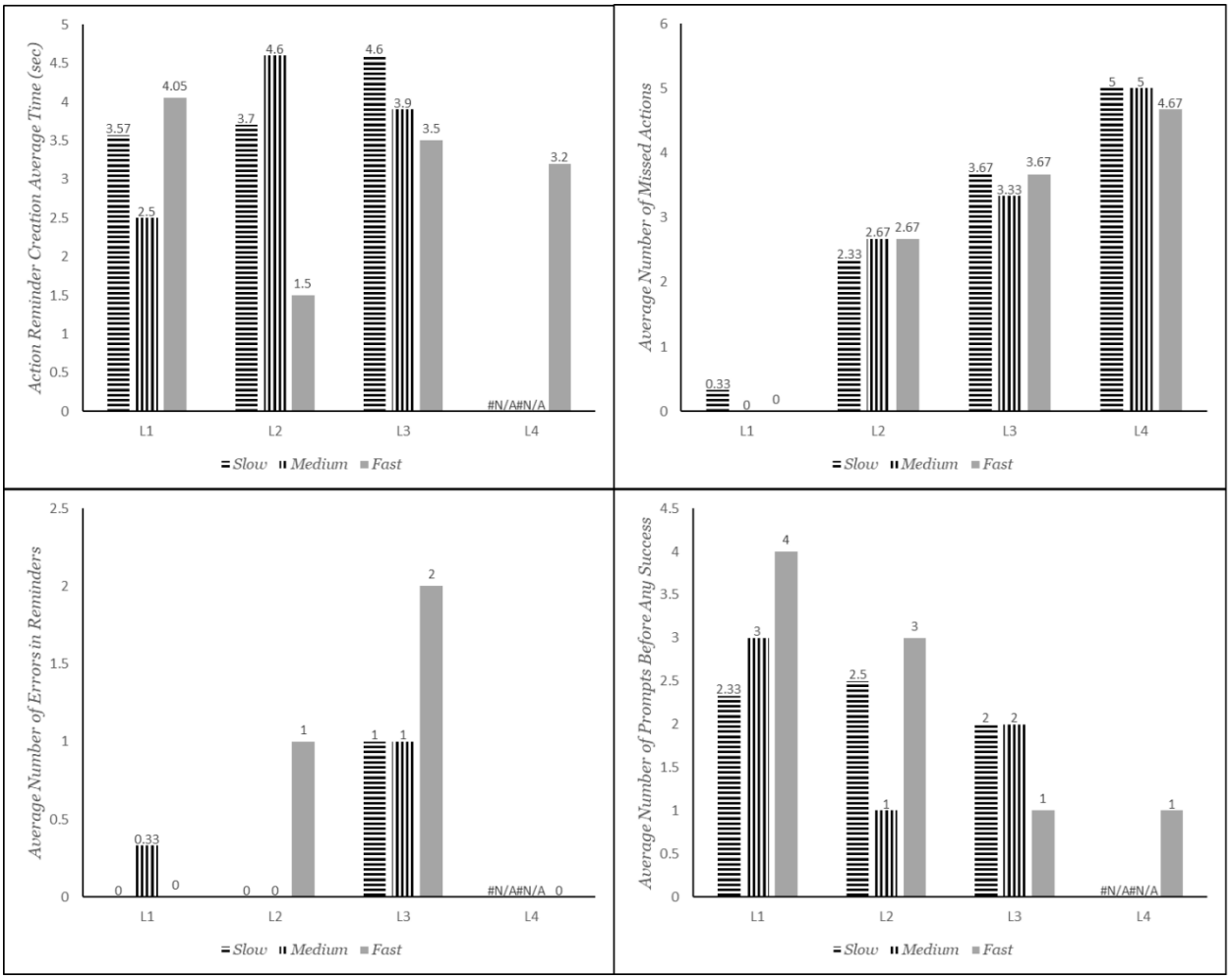

Figure 6. Google Assistant® app performance measurements for each audio sample (L1-L4) and each sample speech speed

Google Gemini® Performance and Observations

Gemini® ARCAT results in Figure 7 indicate a consistent sample processing and reminder creation time between 4.1 sec. and 6.35 sec. in most cases. For slow and medium speed L4 samples, Gemini® timed out suggesting a design choice was made to process only audio input up to a certain length.

ANMA results demonstrate its ability to create multiple action reminders from a single sample. For example, L2 ANMA ranges from 0.33 to 1 (ideal is 0 and worst is 3) which means it was able to create 2 or 3 action reminders out of the desired total of 3. Despite this capability, it had a high number of errors between 1.67 and 2 errors in the reminders created for L2 samples.

An interesting behavior of Gemini® was observed at L3. Overall, it performed better with female-voiced samples, missing fewer reminders while making slightly more mistakes in the created reminders. Another interesting behavior of Gemini® was that it was more likely to miss actions or refuse to create reminders when prompted with slower instruction speeds. In L2 and L3, it performed best at medium instruction speeds. At L4, it performed best at fast instructions speeds. It is possible that Gemini® calculates the complexity (in length) of instructions it receives based on their spoken length rather than word or character count.

ANEIR seemed to follow an interesting trend with no mistakes made in L1, and many mistakes made in L2 compared to its number of tasks. ANEIR for L2, L3, and L4 followed a decreasing pattern with increasing difficulty, most likely due to more actions being missed overall, meaning that there was less room for mistakes to be made.

Gemini® required several prompts on the higher end for reminders to be created. This is again attributed to its tendency to time-out while listening.

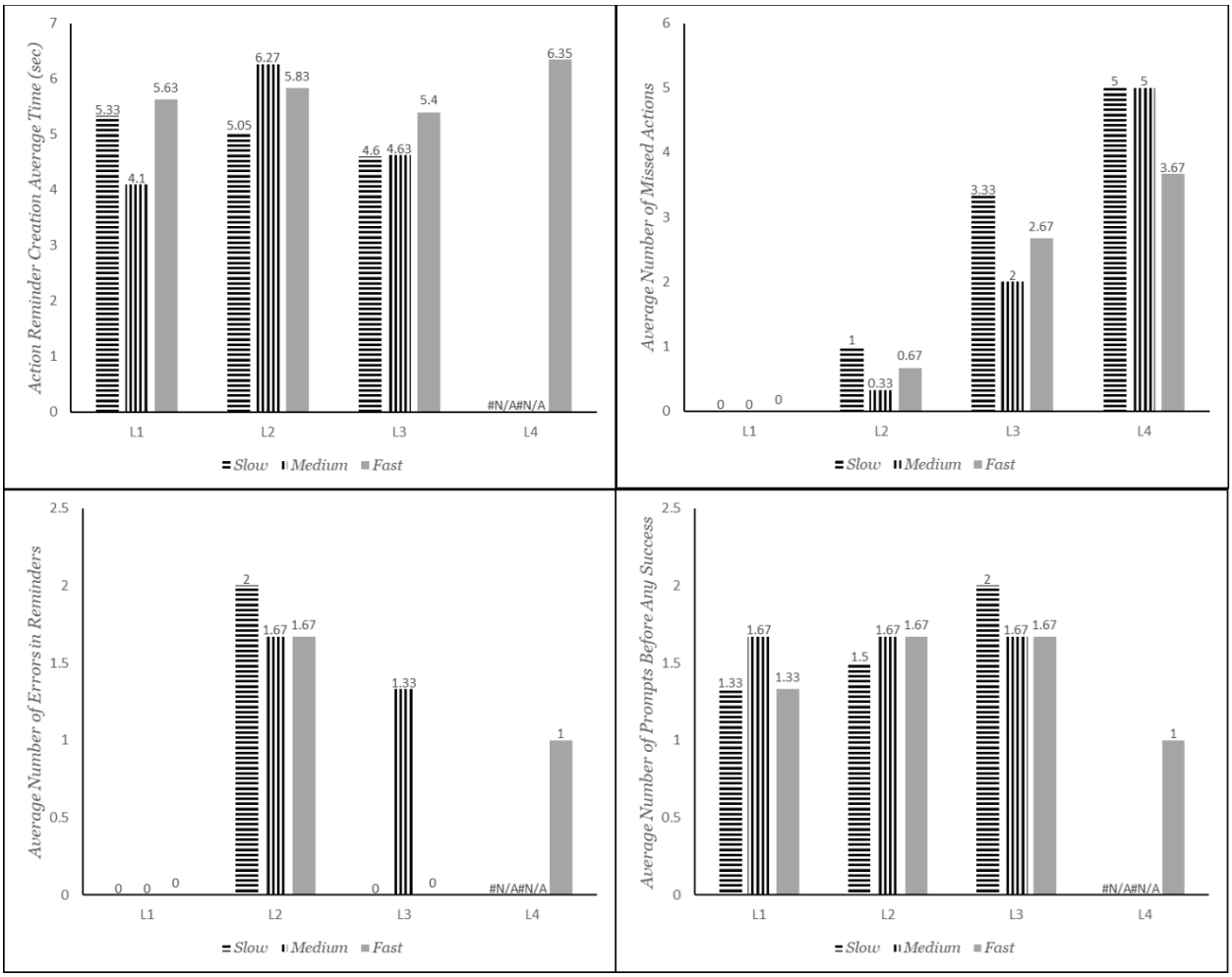

Figure 7. Gemini® app performance measurements for each audio sample (L1-L4) and each sample speech speed

Amazon Alexa® Performance and Observations

Alexa® results are shown in Figure 8. Alexa® processing performance generally exhibited computation time that scaled with complexity from ARCAT results. It is not well-understood why Alexa® needed more time for L1 samples relative to L2. It is possible that L1 fell into what can be considered minimum overhead for speed-to-text or AI engine minimum response time.

It was observed that Alexa® attempts to place the entire sample text in a single reminder; it was incapable of creating multiple reminders from a single sample. While this behavior may suggest that no actions would be missed but errors would be high, ANMA results for L2-L4 indicate that Alexa® did not capture the entire sample and missed one or more actions. Furthermore, the misses increased with complexity and speed.

On average, Alexa® was only able to create reminders without errors for trivial L1 samples demanding one action. Since it was only able to create a maximum of one reminder that incorrectly encompasses all actions, ANEIR results reflect the errors of actions bundled at wrong times. Here, the number of errors also generally increased due to bundling actions in the same reminder as complexity and speed increased. Additionally, for all four difficulty levels, errors were observed the least at medium sample speed (or equal to errors for another speed, but never greater).

Finally, for most cases, Alexa® required about two prompts to create any reminder with a worst case of three prompts needed to act on the fastest and most complex sample.

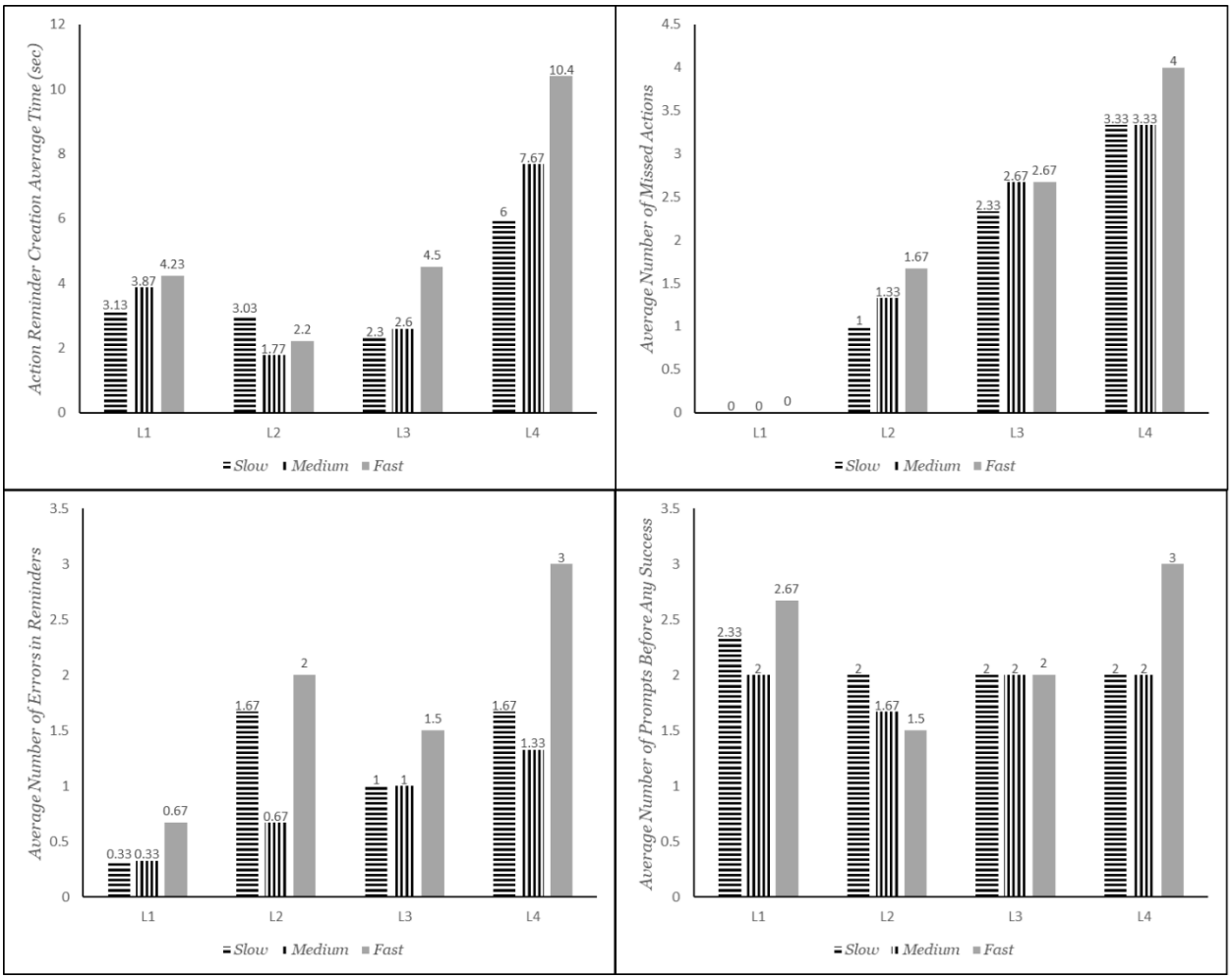

Figure 8. Alexa® app performance measurements for each audio sample (L1-L4) and each sample speech speed

MUNA-GPT3.5T Performance and Observations

MUNA-GPT3.5T results are shown in Figure 9. ARCAT seemed to follow a very slightly increasing trend as difficulty increased, with some outliers such L2 at fast speed and L4 at slow speed. It is possible that these effects are caused by the API's instability. No correlation between sample speed and ARCAT is exposed.

ANMA followed a slightly increasing trend until L4, which was a large increase in misses from L3. The trivial L1 was completed with no actions missed. MUNA-GPT3.5T also performed well overall at L2 and L3, consistently averaging under 1.5 misses for samples that involved 3-4 actions. At L4, ANMA significantly increased, with almost all actions being missed consistently at all sample speeds. It is possible that GPT-3.5T has a processing barrier that lies somewhere between the lengths of the samples for L3 and L4.

ANEIR does not seem to have a correlation with difficulty. At L4, this can be explained by the large number of misses. This leads to mistakes being lower as fewer actions were created, leaving less room for error.

ANPBS remained at a constant one prompt throughout the entire experiment. This is because ChatGPT® was instructed to extract every small detail from each prompt and create reminders from it instead of missing details and asking the user to provide them again. To emphasize reducing the burden on the elderly, MUNA was designed to

never ask the user for more information or to repeat the audio. It creates reminders from exactly one prompt after being given text that contained actions.

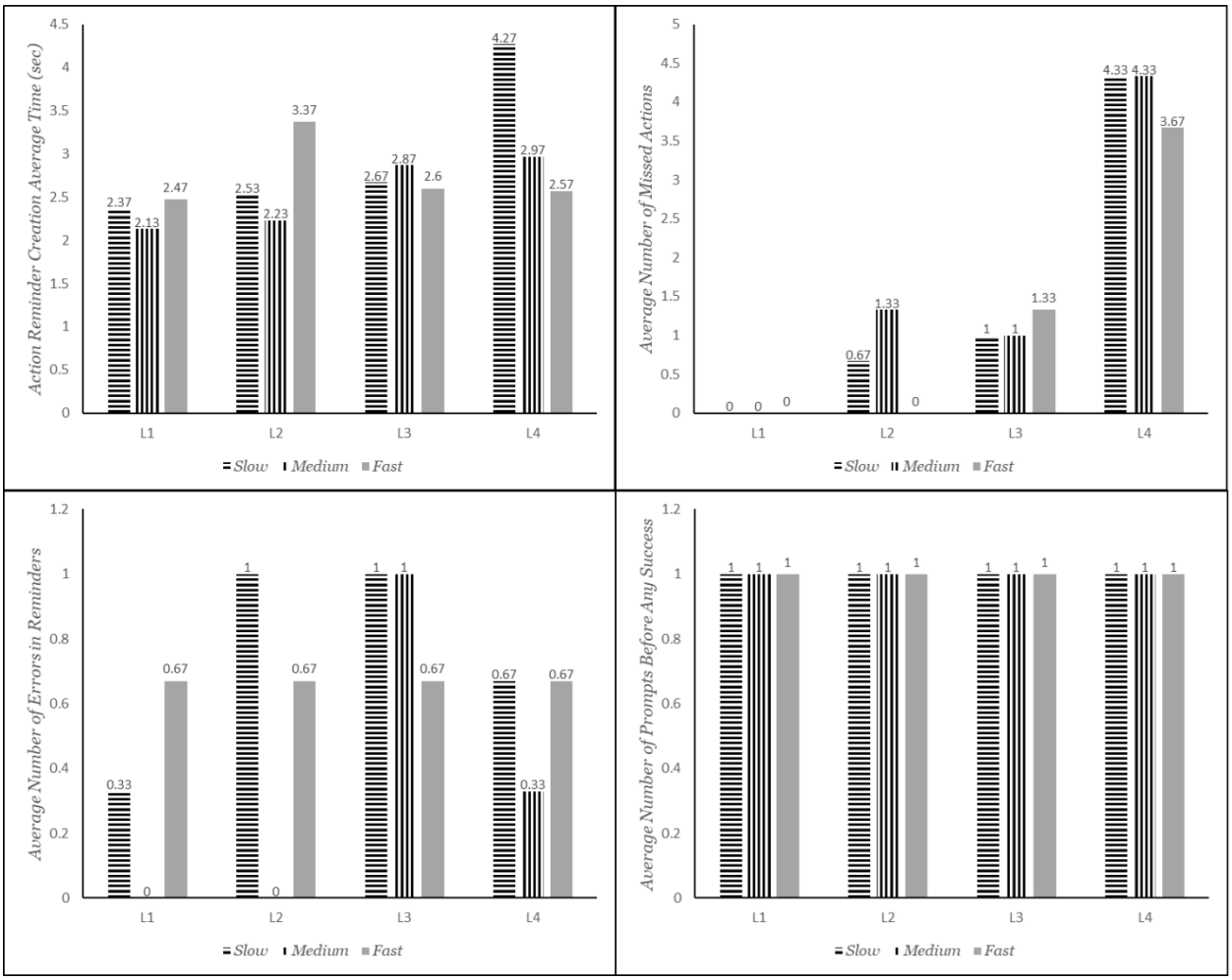

Figure 9. MUNA-GPT3.5T app performance measurements for each audio sample (L1-L4) and each sample speech speed

MUNA-GPT4o Performance and Observations

MUNA-GPT4o results are shown in Figure 10. ARCAT follows a slightly increasing trend, with slow sample speed taking the longest for reminder creation in L1, L2, and L4.

Both ANMA and ANEIR showed zero misses and mistakes at all sample speeds for L1 and L2, meaning the required reminders were created perfectly. Surprisingly, ANMA was overall higher in L3 than in L4, meaning that MUNA-GPT4o missed more tasks overall in L3 even though L3 demanded fewer actions. A possible explanation for this is that the sample for L3 in this graph was prompted in a male voice, with L4 being prompted in a female voice. Since L4 is more complex and requires more reminders than L3, and L3 had fewer misses, this could mean that Android® speech-to-text recognized text more accurately from the female-voiced sample.

Furthermore, when experimenting with L3 in a female voice, there were averages of 0.33, 0.67, and 1.0, for sample speeds slow, medium, and fast respectively. As seen in Figure 10, L3 with the male voice averaged 1.0, 0.67,

and 1.67 for sample speeds slow, medium, and fast respectively. This means that for all L3 speeds prompted in the male voice, ANMA was greater than or equal to the ANMA for all L3 speeds prompted in the female voice.

Performance for L3 and L4 remains strong overall, exhibiting a maximum of 1.67 tasks being missed out of 4 for L3, and a maximum of 1.33 tasks being missed out of 5 for L4.

ANEIR seemed to follow a similar pattern to ANMA, with fewer overall mistakes in L4, possibly for the same reason previously discussed. Interestingly, the only mistakes ever made by MUNA-GPT4o were on medium speeds in L3 and L4.

ANPBS remained constant at one prompt, as MUNA-GPT4o is designed to immediately create calendar reminders upon receiving instructions. It does not prompt the user to provide more information or to check its current information, as this would increase the cognitive load on the user.

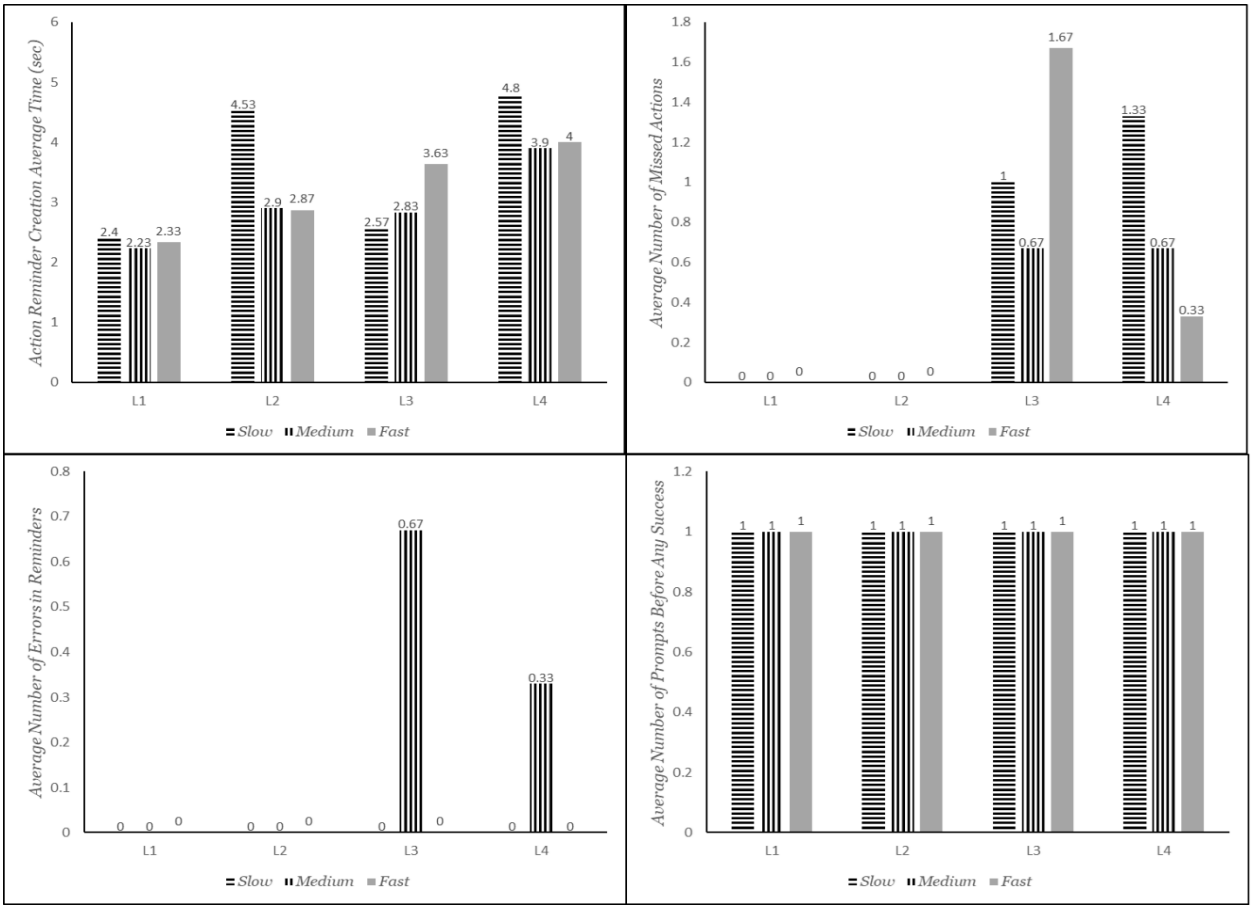

Figure 10. MUNA-GPT4o app performance measurements for each audio sample (L1-L4) and each sample speech speed

Comparative Results and Observations

To understand the performance delta of MUNA-GPT3.5 Turbo and MUNA-GPT4o app prototypes relative to existing assistants, the combined results of ANMA against all assistants are inspected. This is desirable since it can reasonably answer the question: how do the capabilities of transcription, natural language processing, classification, and action extraction compare across all solutions?

Comparing ARCAT, ANEIR, and ANPBS across all assistants is less trivial due to the observation that only Gemini® was found to have the ability to create more than one reminder. For example, comparing ANEIR across all can be misleading since errors in existing assistants other than Gemini® correlate in many cases to their inability to create multiple reminders. Hence, for the rest of the comparative analysis, the focus is to understand the performance of both MUNA flavors relative to only Gemini®.

Average Number of Missed Actions

For L1, the only assistant to miss the single action was Google Assistant® with slow samples as show in Figure 11. This trend also shows up in Gemini®, with the most misses for every difficulty occurring at slow samples. This appears to be a common weakness of both assistants.

For L2, the only assistants that did not miss an action were Siri® at all speeds, MUNA-GPT3.5T in fast samples, and MUNA-GPT4o at all speeds. Siri® did not miss actions because it embedded every word of the sample provided into the name of the reminder. This created many action time attribute errors, but no action reminder creation misses. MUNA-GPT-3.5T and Gemini® performed well overall due to their ability to create multiple reminders at once. The ChatGPT-4o® large processing power and conversational-AI abilities behind MUNA-GPT4o allowed it to complete L2 flawlessly.

MUNA-GPT3.5T and MUNA-GPT4o missed notably less actions than all other assistants at L3, again due to their conversational-AI abilities and reduced resource restrictions for message length and complexity. At slow speed, both MUNA-GPT3.5T and MUNA-GPT4o missed an average of only one action out of four. At this speed, they also had 3.67x less average misses than Google Assistant® and about 3.3x less average misses of Gemini®, and about 2.3x less average misses than Alexa®. Siri® had zero misses due to its strategy to bundle each word of the sample into the reminder title. However, Siri's ANEIR at this data point was the highest.

For L4, Siri® missed all actions in every sample, as it failed to create a single reminder due to the same "for who?" issue described previously. Google Assistant® and Gemini® also performed poorly, with Gemini® slightly improving on faster sample speed. Again, a possibility for this may be that Gemini® measures prompt complexity by spoken time length rather than word or character count. Alexa® consistently outperformed nearly every model except MUNA-GPT4o. This is because Alexa® went about creating complex reminders in a very similar way to Siri®, bundling the sample instructions into the title. However, Alexa® did seem to have slightly more capability than Siri® in this regard, as it avoided confusion or failure more effectively. MUNA-GPT4o performed the absolute best by far in L4, with the highest number of average misses being over 2.5x less than the lowest number of average misses for any other assistant. Interestingly, the misses seemed to decrease as sample speed increased.

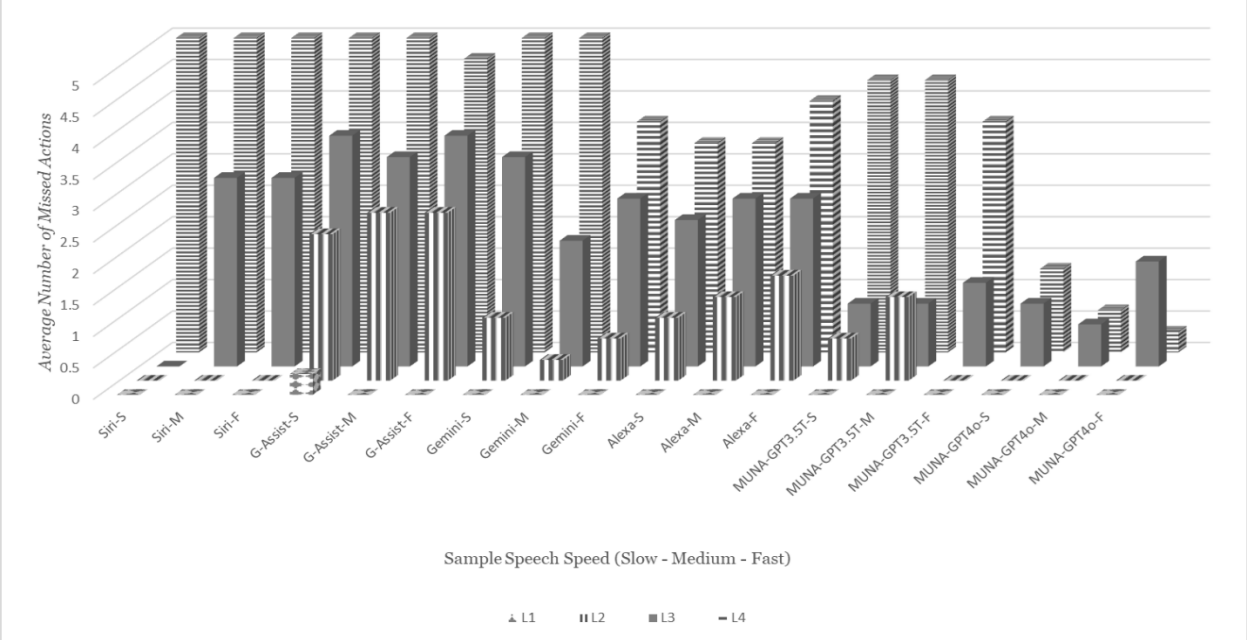

Figure 11. ANMA for all assistants across all samples of complexity and speech speed

Action Reminder Creation Average Time and Average Number of Errors in Reminders

The reminder creation average times are shown in Figure 12 for all conversational-AI models capable of creating multiple reminders at once in this experiment (Gemini®, MUNA-GPT3.5T, MUNA-GPT4o). L4 was not included because Gemini® had too many complete failures, refusing to create a single reminder or timing out.

For L1, MUNA models created reminders with very similar processing speeds. No notable correlation in ARCAT is present amongst each sample speed for MUNA models. At all sample speeds, Gemini® was much slower than both MUNA models, with its maximum ARCAT at fast speed being about 2.3x the maximum ARCAT throughout all MUNA models. At slow and fast sample speeds, Gemini® took notably longer to create reminders than at the medium sample speed.

For L2, Gemini® again created reminders the slowest. Taking every sample speed into account, the highest ARCAT amongst both MUNA models was lower than the lowest ARCAT for Gemini®. This time, however, MUNA-GPT4o performed slower than MUNA-GPT3.5T. A possibility for this result is that GPT-3.5T® is less resource-intensive, resulting in faster processing.

For L3, Gemini® also created reminders slower than both MUNA models at all speeds, with the same statistic discussed for L2 holding true once again. At slow and medium sample speeds, MUNA-GPT3.5T and MUNA-GPT4o had very similar ARCAT levels, but ARCAT was notably higher for MUNA-GPT4o at fast sample speed.

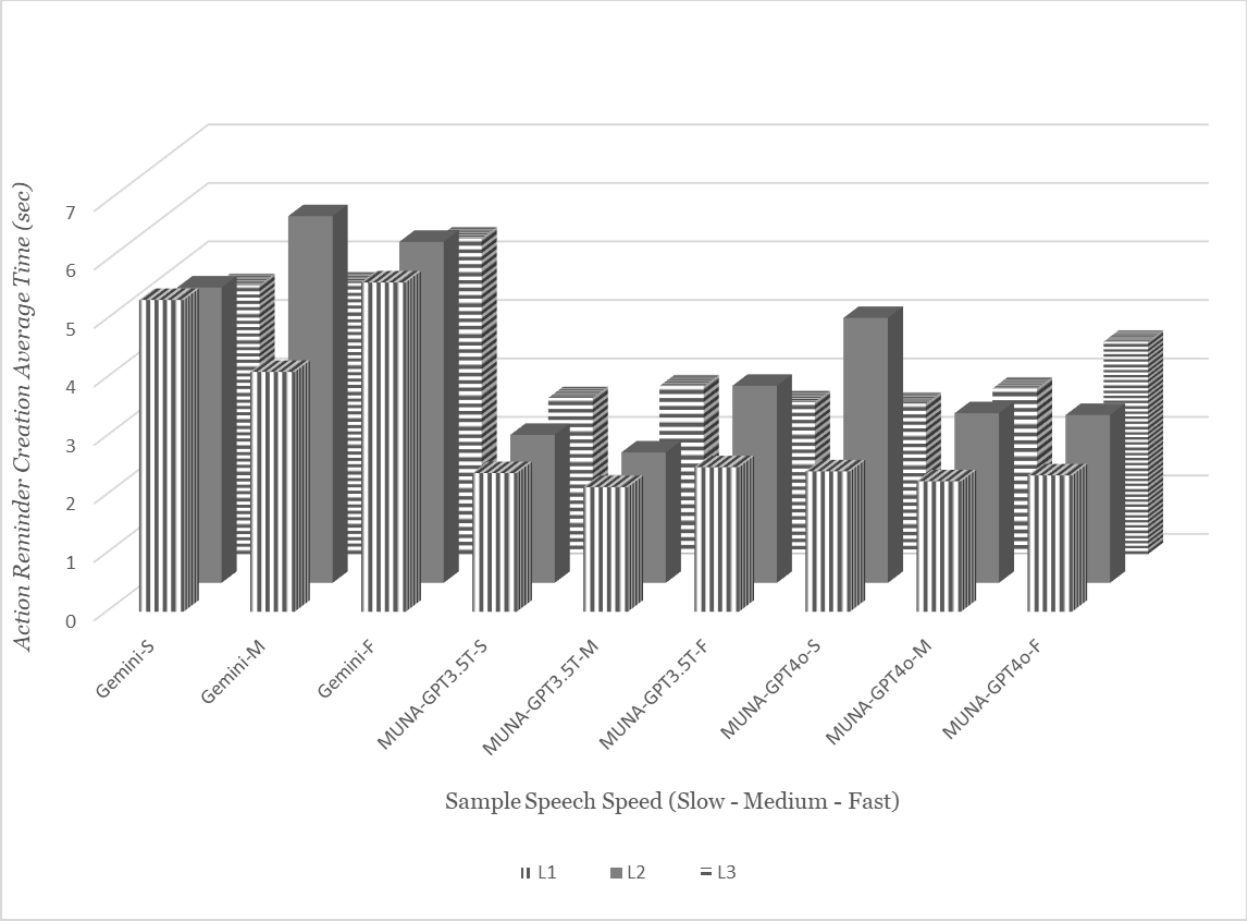

Figure 12. ARCAT for Gemini® and both MUNA flavors across all samples of complexity and speech speeds

The ANEIR chart for all conversational-AI models capable of creating multiple reminders at once (Gemini®, MUNA-GPT3.5T, MUNA-GPT4o) is shown below in Figure 13.

For L1, MUNA-GPT4o and Gemini® both made no errors in any test cases at all sample speeds. For context, they also had zero misses for all these cases, meaning that reminders were created perfectly. MUNA-GPT3.5T made some errors at slow and fast sample speeds.

For L2, MUNA-GPT4o displayed its powerful multi-action processing capabilities, making zero mistakes, and, for context, zero action misses. MUNA-GPT3.5T also made less errors than Gemini® at all speeds, making zero errors at medium speed. At fast sample speed, the ANEIR for MUNA-GPT3.5T was about 2.5x less than the ANEIR for Gemini®.

For L3, MUNA-GPT4o and Gemini® made zero mistakes on slow and fast sample speeds. Interestingly, errors were made at medium sample speeds. MUNA-GPT4o's ANEIR at medium speed was still about 0.5x the same measure for Gemini®. Some important context for these measurements is both MUNA models had fewer action misses than Gemini® at all sample speeds, meaning that they had more room for error and were prone to a higher ANEIR. MUNA-GPT3.5T made the most errors overall, the cause of which is most likely a combination of it creating more reminders than Gemini® overall and having less processing power than MUNA-GPT4o.

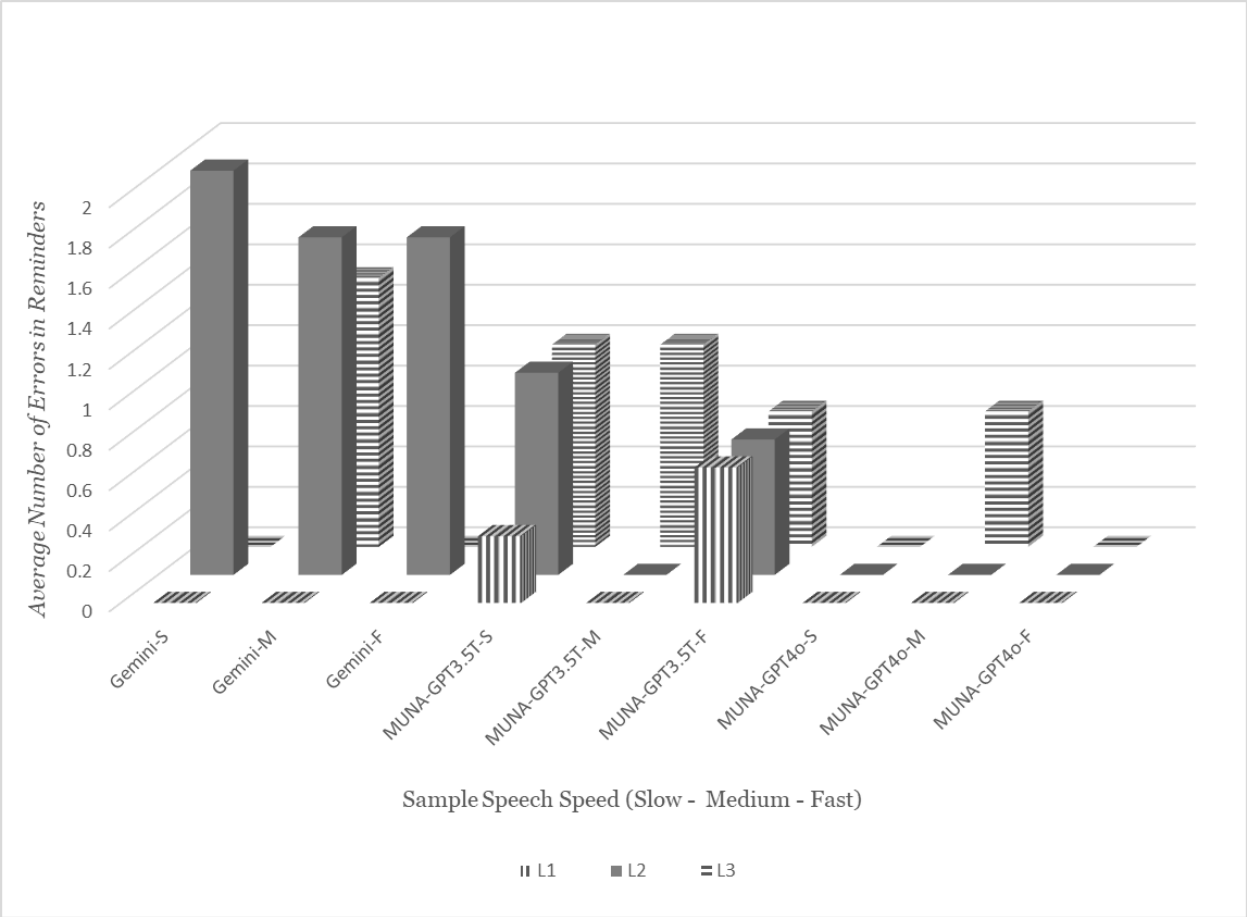

Figure 13. ANEIR for Gemini® and both MUNA flavors across all samples of complexity and speech speeds

Discussion

Assistants Without Conversational-AI

Assistants without conversational-AI (Google Assistant®, Siri®) performed more poorly overall compared to assistants that had conversational-AI capabilities. This measure is mainly reflected in ANMA, where these two assistants missed more action reminders overall on difficult levels such as L3 and L4 than any other assistant in this experiment.

Furthermore, assistants without conversational-AI were confused by the instructions more frequently than conversational-AI assistants. For example, Siri® would constantly ask "for who?" at higher difficulty levels rather than understanding the instructions and creating reminders. Google Assistant® crashed and timed out very frequently, or responded to the pre-prompt and sample with a statement along the lines of "Sorry, I cannot do this". These assistants had the most total failures overall throughout the experiment. However, Siri® did show one strength, as it had a consistent and very short time to create reminders if it understood the pre-prompt and sample it was given.

Assistants with Conversational-AI

Assistants with conversational-AI (Alexa®, Gemini®, MUNA-GPT3.5T, MUNA-GPT4o) performed much better overall compared to assistants with no conversational-AI capabilities. The main reason for this is that these assistants

were much more likely to understand the pre-prompt and prepare to receive instructions. For example, Alexa® and Gemini® responded to the pre-prompt with statements along the lines of "What are the instructions?" or "What reminders should I create?" However, what set Gemini® apart from Alexa® was its ability to create multiple reminders from only one sample. Though, throughout the experiment, Gemini® utilized this capability inconsistently, rarely creating multiple reminders at once without making mistakes in action time or name attributes.

MUNA-GPT3.5T performed exceptionally in the first three levels, yet fell short when provided with the very large prompt and high number of actions for L4. MUNA-GPT4o was similar to MUNA-GPT3.5T yet with much more processing power, allowing it to surpass the resource barrier that impeded GPT-3.5T®, performing exceptionally on L4, and being the only assistant to complete L1 and L2 with zero mistakes and misses. The lowest-performing assistant in the conversational-AI category, Alexa®, still performed better than the best performing assistant in the non-conversational AI category, Siri®.

Conclusion

This research demonstrated that AI capabilities can be deployed effectively on the smart devices of the elderly aging in place to offload the cognitive burden when handling healthcare instructions.

Non-conversational personal assistants were found to be unsuitable for the use case in this research due to their design choice being centered around performing a single request or requests optimized for the smart-home controls they have been commercialized for.

Assistants that use conversational-AI, especially with the advent of GPT4o, are much better suited for this use case as shown through the results of MUNA-GPT4o. The gap between both MUNA versions and the existing conversational assistants tested and the lack of apps similar to MUNA in app stores suggest that there is insufficient attention to helping the elderly compensate for cognitive decline at least in the use case of handling healthcare instructions.

Limitations

Experimental Limitations

- The statistical analysis discussed is reduced by two major limitations. Firstly, the sample size of apps is very small due to the limited number of personal assistants given market domination or consolidation in addition to the lack of competition to such assistants in the app store. Secondly, the behavior of studied apps is not confirmed to follow a particular probability distribution function in response to the population of audio samples. As shown in results discussions, both erratic as well as fixed response types are observed in several cases of existing assistants.
- At the time of performing the experiments with Siri®, Apple Intelligence® was not available. Apple Intelligence® is another conversational-AI, meaning it would have been valuable to compare against both MUNA versions, as the only other assistants that utilized conversational-AI in this experiment were Gemini® and Alexa®.
- Different languages and dialects apart from American English were not tested, even though the backend capability of ChatGPT® can process multiple languages.
- The app capability was not tested with slower Internet speeds or any simulated Internet connectivity issues.

User Limitations

- The prototype does not intend to compensate for any visual or hearing impairments that the elderly may have.

- The prototype also does not compensate for people who require caregivers in their lives.
- Since the app is a prototype and as is the case with consuming results from conversational AI at the time of this research, the user is advised to treat its outputs as experimental.

Future Work

This research did not involve human subjects in testing any existing apps or new app prototypes nor did it involve any subjects in handling the same audio samples present to the apps compared in this research. Further experimentation at a higher education institution with approval from an Institutional Review Board (IRB) would help perform a comparison between the app prototype performance and elderly subjects' performance handling of the same audio samples.

The performance of Siri 2.0® seems to be rapidly improving with Apple® offering users the option to utilize ChatGPT® as a backend. Future updated research experimentation would provide insights into the differences between how effectively MUNA-GPT4o utilizes ChatGPT-4o® vs how Siri 2.0® would.

Improvements of the app prototype towards creating an official product need to address privacy, safety and security concerns. Future work would resolve several open issues around how to isolate each user's ChatGPT® history while using the app. Disclaimers for users to confirm the accuracy of created reminders with a mechanism to report any errors to improve future versions need to be presented and consented for.

Finally, several improvements related to user experience are to be investigated including configurability to choose where summaries are stored, how the app behaves when encountering sample processing issues, and how it forms reminder content including different language choices.

Acknowledgments

I'd like to acknowledge Prof. Reid Simmons and Prof. Tom Mitchell at Carnegie Mellon University for their consulting availability and suggestions on problem areas related to telehealth and healthcare management impacting the elderly aging in place. In the process of exploring a broader set of problem areas, I'd like to thank Prof. Gregory Abowd and Prof. Mathew Yarossi at Northeastern University for their consulting availability on the latest assistive technology research activities at different academic institutions. Finally, I'd like to thank Computer Science Teacher Mr. Gregory Theos at The Advanced Math and Science Academy Charter School for his advice and review of this research.

References

Busch, P. A., Hausvik, G. I., Ropstad, O. K., & Pettersen, D. (2021). Smartphone usage among older adults. *Computers in Human Behavior, 121*. doi: https://doi.org/10.1016/j.chb.2021.106783

C, K., J, F., N, W., E, N. P., S, Z., & M, M. (2020). Utilization Barriers and Medical Outcomes Commensurate With the Use of Telehealth Among Older Adults: Systematic Review. *JMIR Med Inform, 8*(8). doi: https://doi.org/10.2196/20359

Faverio, M. (2022). *Share of those 65 and older who are tech users has grown in the past decade.* Pew Research Center. Retrieved from https://www.pewresearch.org/short-reads/2022/01/13/share-of-those-65-and-older-who-are-tech-users-has-grown-in-the-past-decade/

Iancu, I., & Iancu, B. (2020). Designing mobile technology for elderly. A theoretical overview. *Technological Forecasting and Social Change, 155*. doi: https://doi.org/10.1016/j.techfore.2020.119977

K, N., A, D., K, B., M, K.-U., T, G., & E., S. (2016). Temporal Information Processing and its Relation to Executive Functions in Elderly Individuals. *Frontiers in Psychology*. doi: https://doi.org/10.3389/fpsyg.2016.01599

OtterAI. (2024). Retrieved from https://play.google.com/store/apps/details?id=com.aisense.otter&hl=en_US

P, R., M, W., C, B., S, T., K, S., M, K., & A, M. (2018). Prevalence of Health App Use Among Older Adults in Germany: National Survey. *JMIR Mhealth Uhealth, 6*(1). doi: https://doi.org/10.2196/mhealth.8619

P, W., SJ, B., G, P., I, S., C, T., G, B., . . . S, A. (2017). Using eHealth Technologies: Interests, Preferences, and Concerns of Older Adults. *Interact J Med Res, 6*(1). doi: https://doi.org/10.2196/ijmr.4447

Petersen, R. C., & Negash, S. (2008). Mild Cognitive Impairment: An Overview. *CNS Spectrums, 13*, 45-53. doi: https://doi.org/10.1017/S1092852900016151

Shallice, T. (1982). Specific impairments of planning. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences, 298*, 199-209. doi: https://doi.org/10.1098/rstb.1982.0082

Sharkey, P. (2024). Homebound: The Long-Term Rise in Time Spent at Home Among U.S. Adults. *Sociological Science, 11*(20), 553-578. doi: https://doi.org/10.15195/v11.a20

SoundTypeAI. (2024). Retrieved from https://play.google.com/store/apps/details?id=com.innosq.soundtypeai&hl=en_US

Vockley, M. (2015). The Rise of Telehealth: 'Triple Aim,' Innovative Technology, and Popular Demand Are Spearheading New Models of Health and Wellness Care. *Biomedical Instrumentation & Technology*. doi: https://doi.org/10.2345/0899-8205-49.5.306