TEL AVIV UNIVERSITY

The Lester and Sally Entin Faculty of Humanities
The School of Philosophy, Linguistics and Science Studies
Department of Linguistics

Recovery Strategies in Dialogue with Senior Adults

MA thesis submitted by
Shaul Ashkenazi

Prepared under the supervision of

Prof. Mira Ariel
Tel Aviv University

Prof. Bonnie Webber
University of Edinburgh

Dr. Maria Wolters
University of Edinburgh

October 2021

Abstract

Voice-based assistants, such as Apple's Siri, have become ubiquitous, despite the fact that users may still be misunderstood by the machine. When encountering such an error, the machine can try and "rescue" the conversation, invoking a recovery strategy. Elderly people may face these issues far more often than younger people, as they're not "digital natives", and their expectations with respect to conversing with a computer system may be inaccurate. Of course, this would change as time passes by, and more people will be "digital natives" from a young age.

In this work, I'll start by reviewing past studies in the field of Human-Machine-Interaction and Recovery Strategies. In these studies, the elderly population wasn't taken into account, or the focus was the elderly people but not the recovery strategies. The goal of this research is to bridge this gap, and discover which recovery strategy is better for older people, and whether the optimal recovery strategy constructed for the elderly is different for the one constructed for younger people.

I repeated Bohus and Rudnicky's (2005) Wizard-of-Oz "reservation" experiment, replacing conference rooms with restaurant tables. I used two groups from different age groups of native English speakers, one with 13 participants between the ages of 18 and 30, and the other with 13 participants over 65. They were requested to speak with an automated table reservation system on a website, where 2 different recovery strategies were invoked whenever an error occurred. During this interaction, they performed 6 different scenarios of table reservations, according to specific parameters, using their own words. The software was developed specifically for this study, using the PyDial toolkit, a state-of-the-art spoken dialogue system framework.

In this study, I compared how participants responded to different recovery strategies, highlighting differences between the age groups. I discovered that in scenario 5, when the SDS expected "*peanuts*" which was less immediate for the participants than the word "*nuts*", only one of the strategies could make them rephrase their input to "*peanuts*", and assist them to recover and continue with the task.

Acknowledgments

I wish to thank Prof. Mira Ariel for supporting me in this cross country endeavor, and for her guidance and help. In addition, she taught me a lot about developing and presenting my ideas in front of diverse audiences, which helped me grow as a researcher. I would also like to thank Prof. Bonnie Webber and Dr. Maria Wolters for believing in me, and for their many brilliant ideas, without which this work would have been completely different. Spoken Dialogue Systems aren't as heavily researched in Israel as they are in the UK, and I'm happy I was fortunate to work with highly esteemed supervisors with years of experience in this field.

The PyDial framework isn't trivial to use, to say the least, and I want to thank Dr. Stefan Ultes for answering my many emails, and assisting me in understanding how to use and configure it for my specific domains. In addition, I would like to thank Rotem Meiri for his help and guidance with constructing the back-end of the website.

I want to thank the many participants who volunteered for my online experiment. I'm grateful for their patience and for their time.

I would like to extend my thanks to the staff in the Linguistics department for their interesting and inspiring classes: Evan Gary Cohen, Fred Landman, Aya Meltzer-Asscher, Tal Siloni, and a special thank you to Ruti Zussman.

This work would not have been possible without the love and support of my good friends: Tomer and Alon Scheib, Naama and Lior Berman Amir, Igor and Katya Tubis, Ori Levi Ganani, Noa Winer, Lior Shalom, Yael Shor Michaeli, Vered Orian, Luba Weissmann, Dudu Gottlib, Einav Tubi, Tal Agmon, and Ariella Charvet.

I would also like to thank my parents, Shoshana and Avi Ashkenazi, and my sister Ethy. They've been very supportive throughout all this time, which wasn't easy due to my mother's terminal illness.

Contents

1. Introduction

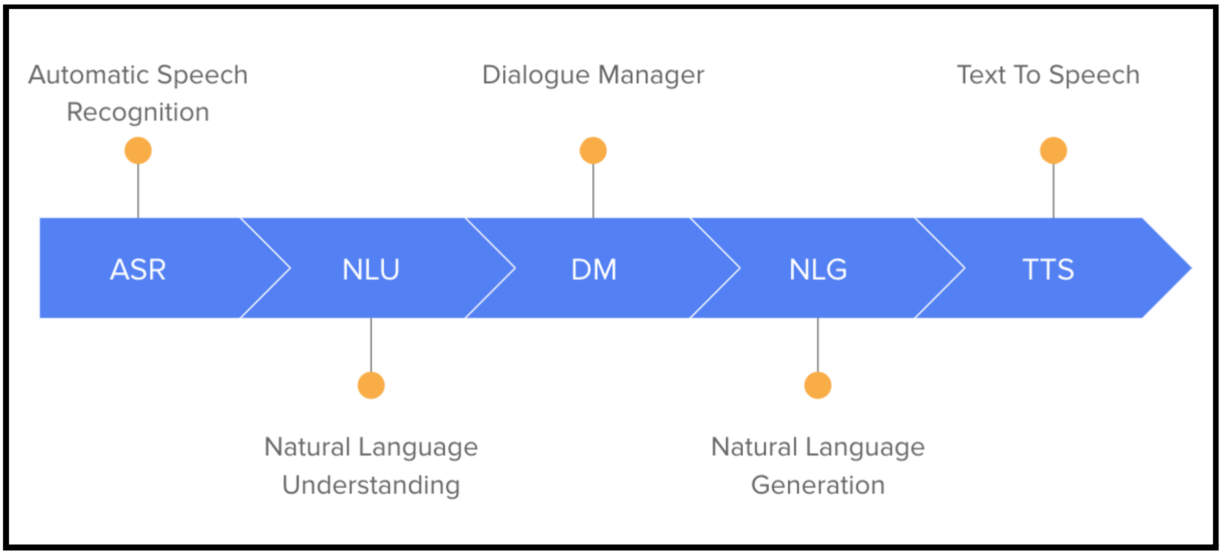
There are now more people using voice based assistants, such as Apple's Siri or Amazon's Alexa, for everyday tasks. These assistants are also referred to as Spoken Dialogue Systems (SDS).

A Spoken Dialogue System consists of 5 components (Figure 1.1):

1. Automatic Speech Recognition (**ASR**): Converts sound waves into text.
2. Natural Language Understanding (**NLU**): Extracts meaning in the relevant context.
3. Dialogue Manager (**DM**): Outputs an instruction for the next response.
4. Natural Language Generation (**NLG**): Generates a response in a natural language.
5. Text To Speech (**TTS**): Generates speech which corresponds to the generated text.

A conversation between a person and an SDS consists of turns, where it's the SDS's job to understand the person, and provide or gather the needed information. When an understanding error occurs during a conversation, the SDS should "understand" that something went wrong, "understand" its cause and then fix it, and prevent the conversation from ending prematurely.

Figure 1.1. The 5 components of the SDS



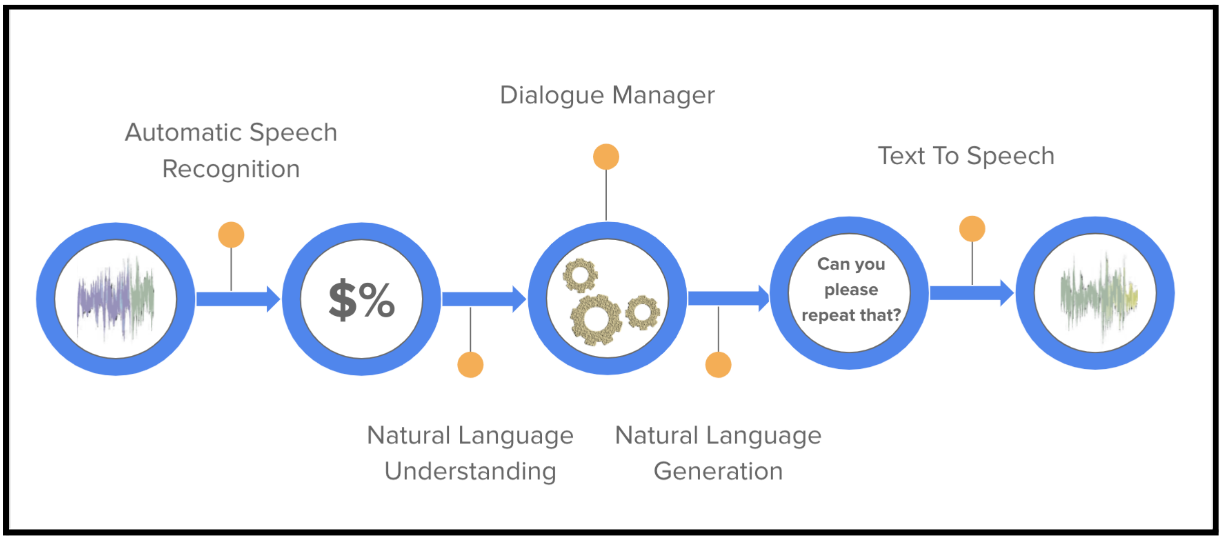
An understanding error can either be a misunderstanding or a non-understanding, where in the former, the user's speech was incorrectly recognized, and in the latter, it was either not interpreted at all, or the recognized speech didn't fit the current state of the conversation or the system's domain. This research focuses only on non-understandings.

These non-understandings can be caused by any component of the SDS. Unlike misunderstandings, the system always knows when it is in a state of a non-understanding. The act of the system responding to such an error, and trying to receive new and correct data is called recovery (Figure 1.2). There are different ways to recover from non-understanding, as we'll see in Chapter 2.

In this research, I compared the performance of two recovery strategies:

- **AskRepeat** - responding with the message: "*Can you please repeat that?*"
- **Reprompt + TerseYouCanSay** - 2-stage strategy:
    - First, responding with a reprompt (repeating the question in a different tone).
    - Following a consecutive error, responding with the message: "Sorry, I didn't catch that. You can Say 'x', 'y' or 'z'", where 'x', 'y' and 'z' are the available options.

Figure 1.2. Recovery from an error



The research questions are:

1. Is one of these recovery strategies better for older people?
2. Is the optimal recovery strategy different for younger and older people?

In chapter 2, I'll review the literature done in the field of recovery strategies, with regard to the elderly population. In chapter 3, I'll present the experiment, and the methods used to construct it. In chapter 4, I'll review the results, followed by a discussion in chapter 5. In the final chapter 6, I'll summarize and show my conclusions.

2. Literature Review

2.1. Wizard of Oz Simulations

When designing a human-computer interface, it’s hard to anticipate the different users’ reactions. Dahlbäck et al. (1993) presented a method of simulating a system involving an AI agent. Instead of developing a fully operating system, testing it, and analyzing where users are experiencing problems with its operation, they used a live person pretending to be the AI agent. This is similar to the “Wonderful Wizard of Oz” story, where Dorothy and her friends interacted with visuals and sounds appearing before them, while the actual wizard of Oz was hiding behind a curtain, pressing buttons and deceiving them.

The motivation behind this pretense is that people adapt to their dialogue partner, and because of that, there will be a language difference between human dialogues and human-computer dialogues. Using this method, it is possible to observe the users using the system, analyze their dialogues, and contribute to the development of the final system.

This is an important technique when exploring recovery strategies, which are sets of system messages that would help the user to stay in the dialogue, and move forward.

2.2. Recovery Strategies

Since the late 20th century, there have been a multitude of studies in the field of error handling and recovery strategies. In this section, I’ll review only a handful of them.

Skantze (2003) thought the Wizard of Oz (WoZ) simulations were perfect for data collection, but they didn't provide data on speech recognition errors. He devised an experiment, where the human wizard could only read ASR processed responses, and not hear the subject's speech, in order to simulate real-time non-understandings. In most cases of non-understandings, the wizards didn't explicitly signal a problem, but instead, they asked the subject a task-related question, or continued the dialogue with a partial understanding. The results showed that signaling of an understanding error (similar to **AskRepeat** in Table 2.1), was the least successful strategy of the 3 methodologies used.

Table 2.1. Selected Recovery Strategies Examined by Bohus and Rudnicky

AskRepeat	*Can you please repeat that?*
Reprompt	[system repeats the previous prompt]
MoveOn	*Sorry, I didn't catch that. One choice would be Wean Hall 7220. This room can accommodate 20 people and has a whiteboard and a projector. Would you like a reservation for this room?*
TerseYouCanSay	*Sorry, I didn't catch that. You can say 'I want a small room' or 'I want a large room'. If the size of the room doesn't matter to you, just say 'I don't care'.*

Bohus and Rudnicky (2005) studied the origins of non-understandings, and came to the conclusion that users who are non-native speakers, first time users, etc. are more prone to experience them. In addition, they suggested 10 different strategies for handling non-understanding errors. e.g., in the strategy they named **MoveOn** (Table 2.1), the system advances the task by moving on to a different question. This specific strategy proved to have the highest performance out of the 10 proposed strategies, and so they got to the same conclusion Skantze arrived at, that it's better to "move on" than to signal an understanding error.

Henderson et al. (2011) also explored recovery strategies in dialogues. In their work, they separated between "*goal-driven*" and "*conversational*" dialogues, where they focused on the latter type. The user interacted with a robot tour guide, asking open questions about items in an exhibition. The researchers used 4 different strategies, with one called "*fake*", where the wizard pretends to add information which was forgotten, instead of signaling a non-understanding. They found that a use of mixed strategies had better performance than the use of the baseline method of "*can you please repeat that?*" (similar to **AskRepeat**).

Using a Wizard of Oz simulation, Opfermann and Pitsch (2017) investigated the interactional impact of three-fold reprompts in dialogues on user actions, where a reprompt is a repeat of the previous system prompt (the same as **Reprompt** in Table 1). This strategy displays non-understanding on a <u>pragmatic level</u> compared to a <u>verbally explicit</u> notification such as "*Sorry, I didn't catch that*". In addition, a reprompt doesn't identify any specific trouble source, and leaves it to the user to decide how to produce her response differently. As opposed to Skantze (2003) and Bohus and Rudniky (2005), who mostly tested undergraduates in their experiments, Opfermann and Pitsch used 3 groups: elderly people

(SEN), people with mild cognitive impairments (CIM), and a control group of undergraduates (CTL). They found that CTLs produced by far more one-word turns than SENs and CIMs, with CIMs showing a decline with each issued reprompt. They concluded that reprompts are efficient if used once, but not multiple times. In addition, they proposed that after the first reprompt, followed by a user reaction, it would be best to use a "you can say" move (**TerseYouCanSay** in Table 2.1).

Kim et al. (2019) investigated error recovery from non-understanding errors with drivers interacting with the in-vehicle voice user interface (VUI). In their WoZ experiment, 47 participants (average age of 25.7 years) were asked to perform driving tasks using a driving simulator. They compared the performance of 3 error recovery strategies: "*ask repeat*" (**AskRepeat**) , "*re-prompt*" (**Reprompt**), and "*you can say*" (similar to **TerseYouCanSay**). The recovery strategies were invoked by the human wizard. The participants were asked to rank the recovery strategies after the experiment was over, and they ranked **Reprompt** as the worst. The researchers agreed with them and concluded that **Reprompt** is the least suitable for drivers, because the participants thought that the VUI didn't hear them at all due to noise or other reasons. The participants favoured **AskRepeat** because it helped to signal that there was a miscommunication. They also favoured the "*you can say*" strategy, and ranked it as the best, although they reported it was difficult for them to remember the example the system used.

2.3. Spoken Dialogue Systems and the Elderly Population

The following have been proposed as the main reasons for the difficulties senior adults are having using SDSs.

Speech recognition

The Automatic Speech Recognition (ASR) component of the SDS, has significantly higher Word Error Rates (WER) for elderly voices than non-elderly adult voices (Vipperla et al., 2008). In other words, it is difficult for the SDS to understand speech produced by elderly people.

Hearing loss

According to the literature review provided by Roth et al. (2011), in Europe, by the age of 70, approximately a quarter of the population was found to have a hearing loss of 30 dB HL (Decibel Hearing Level) or more (see Table 2.2. for detailed hearing loss levels).

Table 2.2. Standardized Hearing Loss Categories (Roth et al., 2011)

Categorization	**'EU' Classification**	**'WHO' Classification**
Normal	dB HL < 21	dB HL < 26
Mild	21 ≤ dB HL < 39	26 ≤ dB HL < 40
Moderate	40 ≤ dB HL < 69	41 ≤ dB HL < 60
Severe	70 ≤ dB HL < 94	61 ≤ dB HL < 80
Profound	94 < dB HL	80 < dB HL

Cognitive decline

Brookmeyer et al. (2007) showed in their article that dementia incidence increases exponentially with age. In addition, they predicted that by the year 2050, 1 out of 85 people will be living with Alzheimer's disease. This cognitive decline makes it difficult for this population to keep a high level of concentration during a long conversation.

Understanding these challenges and trying to tackle the cognitive decline problem, Wolters et. al (2009b) explored strategies they hoped will help reduce the working memory load for elderly users. The strategies they evaluated were presenting fewer options per turn and providing confirmations. They didn't find that reducing the number of options had any effect on task performance. In another study, Wolters et. al (2009a) analyzed 447 appointment scheduling dialogs between 50 older and younger users and 9 simulated SDSs. They distinguished between two groups of users, which they named "*social*" and "*factual*". The "*social*" users chatted with the SDS, and treated it like a human, where the "*factual*" users were more efficient and stayed within the domain limits. The majority of the "*social*" users were, as expected, older users, but on the other hand, over a third were "*factual*" users, making it hard to predict the user's behavior based on age alone.

Kopp et al. (2018) investigated ways in which conversational agents can provide cognitive and emotional assistance to elderly users or people with cognitive impairments. They stated that existing SDSs do not cater to the needs of these kinds of users. In their experiment, they used a specific virtual assistant which was developed to enable "*socially cooperative dialogue"* (*a dialogue style that enables robust and reliable, yet acceptable spoken-language interactions with these user groups*). In order to achieve this, they allowed the user to talk out of turn, make long utterances. The SDS would interrupt them if the utterances were too long. Based on Opfermann and Pitsch's (2017) further research suggestions (see section 2.2), they also used reprompts, followed by clarification messages to understand the user's intentions. In their experiment, they asked participants to enter free-form appointments into a calendar. Similar to Opfermann and Pitsch (2017), they used 3 groups: elderly people (SEN), people with mild cognitive impairments (CIM), and a control group of undergraduates (CTL). They reported that all the participants managed to enter the required number of appointments.

Kobayashi et al. (2019) conducted a study where 40 healthy elderly participants performed tasks on an SDS. The participants started with a practice scenario, followed by 3 tasks: asking for tomorrow's weather, booking a movie ticket, and creating a calendar event. Out of the recorded voices, the researchers extracted 4 vocal features: *pause*, *hesitation*, *delay* and *interruption*. The experiment was a WoZ experiment with a human wizard and simulated error situations. The participants were also evaluated according to their failure to rephrase or correct themselves after a simulated error (behavioural features). The researchers collected cognitive measures of neuropsychological assessments in order to evaluate the cognitive functioning of the participants. They found that the failure in correcting, as well as the failure in rephrasing, increases with cognitive decline. They also found that participants with higher cognitive scores tended to exhibit fewer pauses or hesitations.

3. Methods

3.1. PyDial Toolkit

Bohus and Rudnicky (2005) used the Olympus framework for their study, so for the purpose of replicating their experiment with older participants, using that framework seemed like a good choice for this study, as well. On the other hand, almost two decades had passed, and a lot had changed in the world of SDSs. In his PhD thesis, Olaso Fernández (2017) dedicated an entire chapter for an overview of most of the existing SDS toolkits and frameworks. The most suitable framework for his research was the Olympus, and he used it to create a Spanish SDS.

Olympus can only work on Windows OS systems, and although it is still maintained, its last stable version is from 2015, and each of its components needs different requirements and installations. At the time of Olaso Fernández's research, PyDial didn't exist yet, and so it wasn't covered. When comparing it to Olympus and the rest of the toolkits, I decided it was the most suitable for this research. It was coded in Python, easy to use and to install, and its last stable version is from 2020.

PyDial (Ultes et. al, 2017) is an open source end-to-end statistical SDS toolkit which provides implementations of statistical approaches for all dialogue system modules. It is actively used for research and maintained by the Dialogue Systems and Machine Learning Group at the Heinrich-Heine University, Düsseldorf. It was originally developed by the Dialogue Systems Group at Cambridge University Engineering Department (CUED).

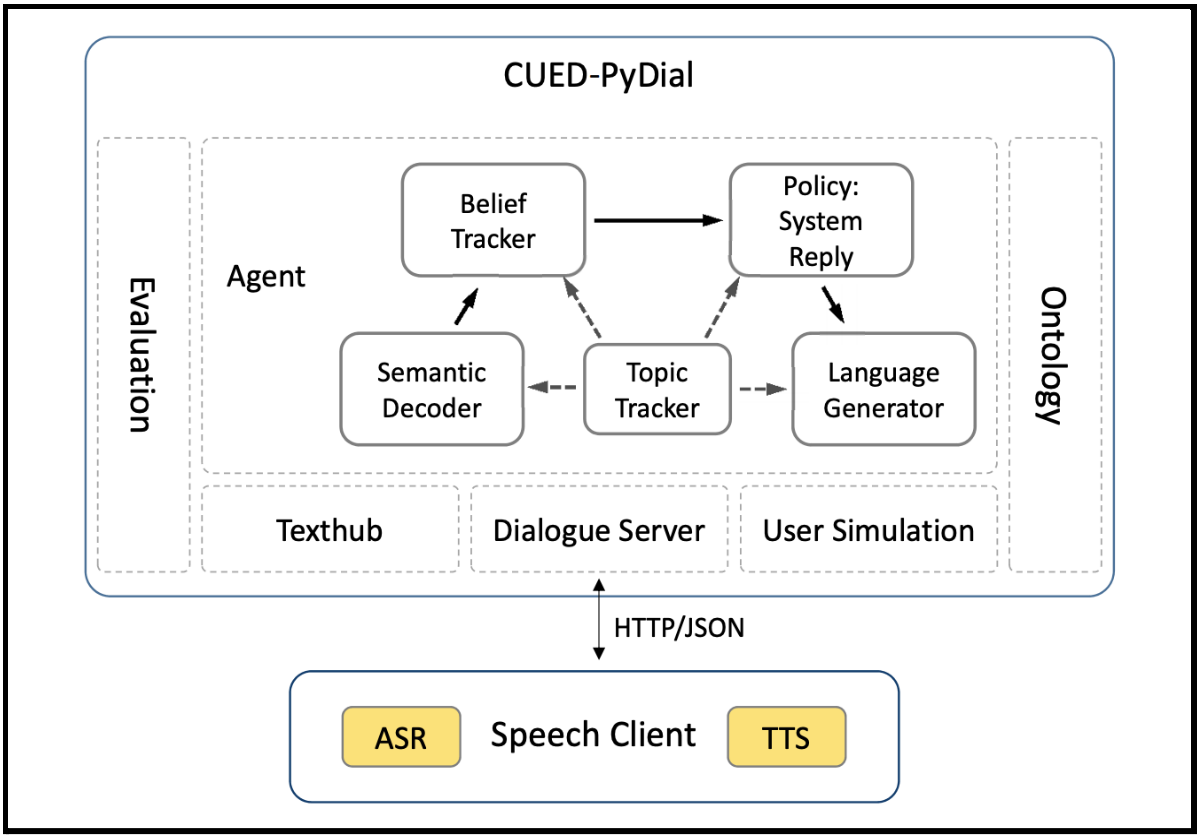
Architecture

The PyDial architecture (see Figure 3.1) has many functionalities, but in this paper, I'll only describe the parts which are relevant to the experiment. It is based on a dialogue agent which queries the user concerning different parameters related to a specified domain. Models for semantic input decoding and language generation of the output are, of course, domain specific. The input and output can be textual or vocal, depending on the client side using the architecture.

The DM component incorporates the policy and the belief tracker (see Figure 3.1), and based on the current belief state, it generates the next dialogue act. In my WoZ experiment, I used the wizard policy (more about this policy at the end of the section), which ignores the belief states, and the next dialogue act is chosen by the wizard.

The domain of the world where the dialogue takes place is configured using the Ontology model. Using the interface the Ontology provides, the dialogue agent can receive data such as slots, slot values from a predefined database. The architecture is very modular and general, in order to keep the domain specific parameters disconnected from the implementation of the different SDS components.

Figure 3.1. PyDial Architecture (Ultes et. al, 2017)



Domain Configuration

I created two domains for the experiment:

1. **PreTest**: The pretest domain for the conversation, which is a preliminary test to screen participants.

2. **RestTables**: The table reservation domain for the experiment.

For each domain, I added an SQLite database and a JSON file, specifying the different slots that are used, and all the valid possible values (see Tables 3.1 and 3.3). Both databases contain a "*name*" column, because although the architecture is domain independent, its main aim is to help a user to find a unique entry in a database, whether it's a restaurant in the campus, or a hotel in the city centre. The unique identifier for such an entry is its name, and since it's hard-coded, I had to use it, as well.

The PreTest database contains only a single entry with fillers, because PyDial doesn't work with an empty database. The reason it only contains fillers, is that there wasn't any query for predefined data, and all the prerecorded messages were almost identical between the participants, except for the recovery strategy that was used. The user is asked about 4 parameters: age, country, day and restaurant (see Table 3.2).

Table 3.1. PreTest Database

name	**day**	**age**	**restaurant**	**country**
1	day	age	restaurant	country

Table 3.2. PreTest Parameters

parameter	**Semantic Output**
day	What day is it today?
age	How old are you?
restaurant	What's the name of your favourite restaurant?
country	Which country are you from?
	Sorry, I didn't catch that. You can say the name of the city you are from.

The RestTable database contains 6 entries corresponding to the 6 scenarios used in the experiment. If the participants would try to give different information, or mix between the scenarios, they would fail to reserve a table in the restaurant. The user is asked about 5 parameters: date, time, size, restriction and phone (see Table 3.4). The size parameter has 3 possible values: **Small** (up to 5 people), **Medium** (between 6 and 10 people) and **Large** (more than 10 people). The task parameter is an addition to the hello message, and it's only a filler, signalling to the user that the task is to reserve a table in case of an error. The phone parameter is an attention parameter, and isn't essential to complete the task successfully.

Table 3.3. RestTables Database

name	task	date	time	size	restriction	phone
1	task	21-11	19:00	medium	kosher	phone
2	task	21-06	20:00	large	vegetarian	phone
3	task	31-08	15:00	medium	lactose	phone
4	task	03-03	12:00	small	halal	phone
5	task	17-09	10:00	small	peanuts	phone
6	task	08-05	9:00	large	gluten	phone

Table 3.4. RestTables Parameters

parameter	Semantic Output
task	Would you like to reserve a table?
date	For which date?
	Sorry, I didn't catch that. You can say the name of the month, followed by a number. For example, November 21.
time	At what time would you like it for?
	Sorry, I didn't catch that. You can say the hour, followed by AM or PM. For example, 8 PM.
size	How many people is it for?
	Sorry, I didn't catch that. You can say the number, followed by "people". For example, 8 people.

restriction	Are there any dietary restrictions?
	Sorry, I didn't catch that. You can say *gluten free*, *halal*, *kosher*, *lactose free*, *peanut free*, *vegetarian*, or *no restrictions*.
phone	Can I take your name and phone number, please?
	Sorry, I didn't catch that. You can say the 7 digits slowly, digit by digit.

The architecture is configurable using a configuration file, in which all the customized models are linked to the domain and its pipeline. After PyDial loads this file, the conversation with the user is ready to start.

Additional System Actions

I mentioned that architecture's aim is to help a user to find a unique entry in a database, and it has a specific set of actions regarding possible slot, such as the followings:

- **request + slot**: Request information in order to fill a slot.
- **repeat**: Repeat the previous system message.

I added implementation and altered the code in a Policy related module named SummaryAction and added additional actions:

- **inform_bye**: Inform the user that there's a table available according to the parameters they specified.
- **quit**: Inform the user that the experiment is over.
- Recovery Strategies:
    - **askrepeat**: Asks the user if they can repeat what they said.
    - **reprompt**: The same as repeat, only with a different name.
    - **tersesay + slot**: Request information in order to fill a slot, while adding relevant options, and offer an example for a valid answer.

WIzard Policy

The architecture had a built-in implementation for a wizard policy. I created a new class which inherited from the PyDial wizard class, in order to have a wizard interface which was more tailored to the experiment needs.

The wizard has to act as quickly as possible, so I added the last system message, the last user input and data about the filled slots. The menu contains only relevant actions (see Figures 3.2 and 3.3).

Figure 3.2. Wizard Policy for PreTest Domain

```
PYDIAL >  Sorry, I didn' t catch that. You can say the name of the city you are from.
USER   >  ('Tel Aviv', 0.9632012)
 1: request_day                          —> request(day)
 2: request_age                          —> request(age)
 3: request_restaurant                   —> request(restaurant)
 4: request_country                      —> request(country)
 5: tersesay_country                     —> tersesay(slot="country")
 6: inform_bye                           —> inform_bye(name="6")
 7: bye                                  —> bye()
 8: askrepeat                            —> askrepeat()
 9: reprompt                             —> tersesay(slot="country")
10: quit                                 —> quit()

Enter the number of the action: 7
```

Figure 3.3. Wizard Policy for RestTables Domain

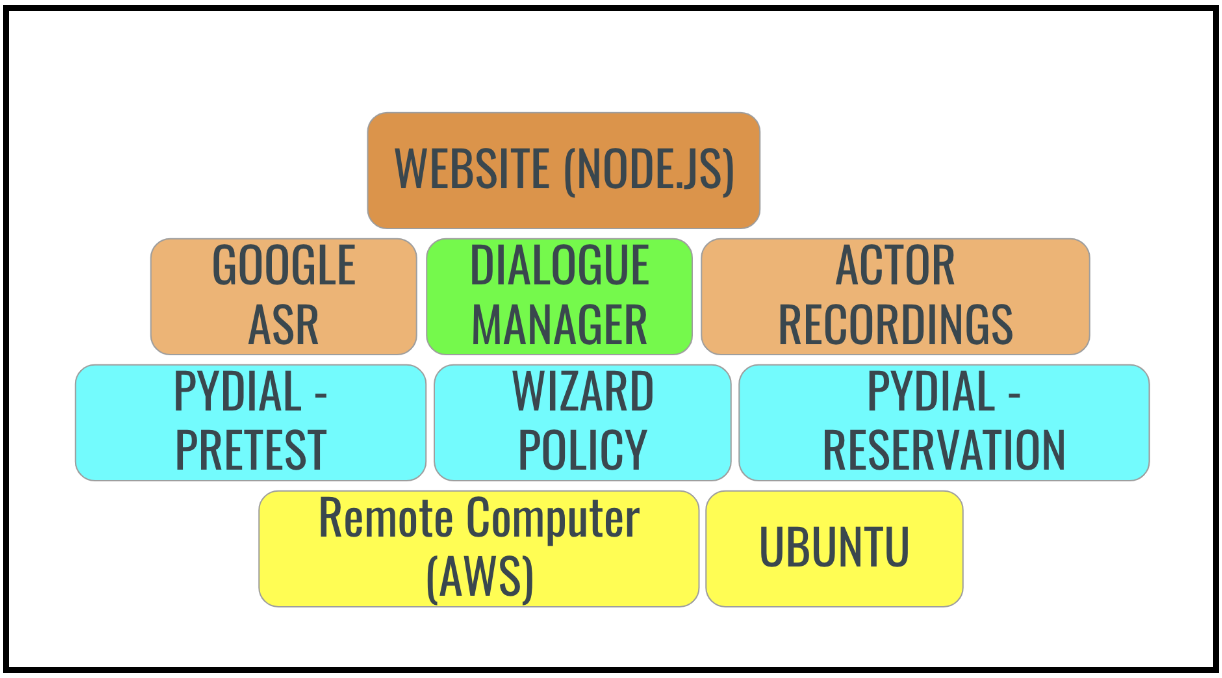
```
PYDIAL >  At what time would you like it for?
USER   >  ("we'd like to come around 8", 0.9770443)
SLOTS:  [ date(21-06) | time(**NONE**) | size(**NONE**) | restriction(**NONE**) | phone(**NONE**) ]
 1: request_task                         --> request(task)
 2: tersesay_task                        --> tersesay(slot="task")
 3: request_date                         --> request(date)
 4: tersesay_date                        --> tersesay(slot="date")
 5: request_time                         --> request(time)
 6: tersesay_time                        --> tersesay(slot="time")
 7: request_size                         --> request(size)
 8: tersesay_size                        --> tersesay(slot="size")
 9: request_restriction                  --> request(restriction)
10: tersesay_restriction                 --> tersesay(slot="restriction", option1="gluten", option2="halal", option3="k
osher", option4="lactose", option5="peanuts", option6="vegetarian")
11: request_phone                        --> request(phone)
12: tersesay_phone                       --> tersesay(slot="phone")
13: inform_bye                           --> inform_bye(name="865",date="21-06")
14: bye                                  --> bye()
15: askrepeat                            --> askrepeat()
16: reprompt                             --> request(time)
17: quit                                 --> quit()
```

In order to add access to the last system message and the last user message, I had to add code in the DialogueServer module, which is the module responsible for the entire conversation pipeline. I accessed the current Policy module from the DialogueServer and saved these two messages as two new properties. During the conversation, this alteration made it possible for the wizard policy to retrieve the message and display them on the screen on the server side.

After each user's vocal input, their voice is recognized as text, and based on that text, the wizard can choose the next system message. In addition, I've added a visual indication for the current status of the slots, and which value do they currently hold (see Figure 3.2). This was done in order to give the wizard visual cues which will help them to make quicker decisions concerning the generation of the next system message.

The wizard can only use actions which ask for information from the user. For the pretest, there aren't any slots which need to be filled, so the wizard decides whether the conversation is a success or not. Concerning the table reservation scenarios, if the user fails to deliver the correct information, in order for the slots to be filled, then the scenario is a failure, regardless of the wizard's desires.

Figure 3.4. Software Architecture



3.2. Online Experiment Architecture

The architecture for the experiment described in section 3.4 consists of 4 layers (see Figure 3.4.), and I'll describe them from bottom up:

1. Remote Linux Machine:

   An Ubuntu machine hosted at Amazon Web Services (AWS).

2. SDS framework:

   I installed the PyDial toolkit and configured two domains: PreTest and RestTables. Both of the domains run separately as two independent PyDial servers, waiting for a conversation to start. After each user input, the wizard was presented with a menu showing different system responses to choose from (see Figures 3.2 and 3.3).

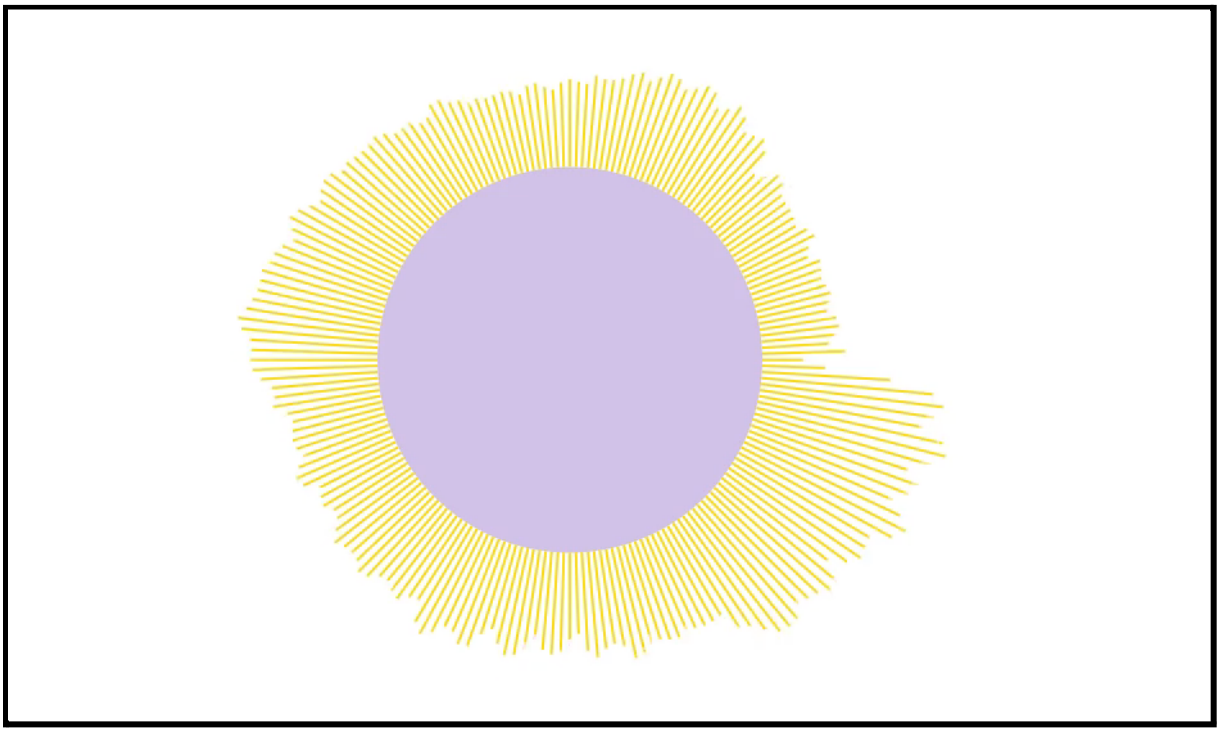
3. Communication with the SDS:
   - Dialogue Manager: This component is responsible for the full integration of the SDS with the pretest and the experiment scenarios together with the ASR and the TTS. I've coded it especially for this experiment:
     - Client side: It connects to the PreTest PyDial server, and after the pretest is over, goes through at most 6 iterations of connecting with the RestTables PyDial server. It could be less than 6 iterations if the wizard notices that the participant is in distress, or that the experiment needs to be over sooner than expected.
     - ASR & TTS: It waits for input recordings to be received from the website, in order to send them for the ASR, and generates HTML pages with the path to the next pre-recorded system message, to be displayed by the website.
     - Logging: It creates a transcription for the conversation with timestamps, and stores the recordings of each scenario in a different folder, for later use and analysis.
   - ASR: The voice recorded by the user was sent from their computer to the remote linux machine from the first layer. The recording was converted into text using the Google Cloud speech-to-text online service.
   - TTS: All the textual messages generated by the SDS were pre-recorded by a male native English speaking actor. A male voice was used, because previous

research conducted by researchers such as Mihailidis et al. (2008) suggests that male voices are easier to hear and understand, possibly because the male voice has a lower pitch/frequency.

4. Website:

   I’ve created a website using node.js, an open-source, cross-platform, back-end JavaScript runtime environment. This website shows html pages generated by the dialogue manager, recording input from the user and playing pre-recorded messages from the SDS. When the recording is played, the user sees a vibrating sun on the screen, which helps to create the illusion of speaking with a virtual agent (see Figure 3.5).

Figure 3.5. Visual indication of an SDS message

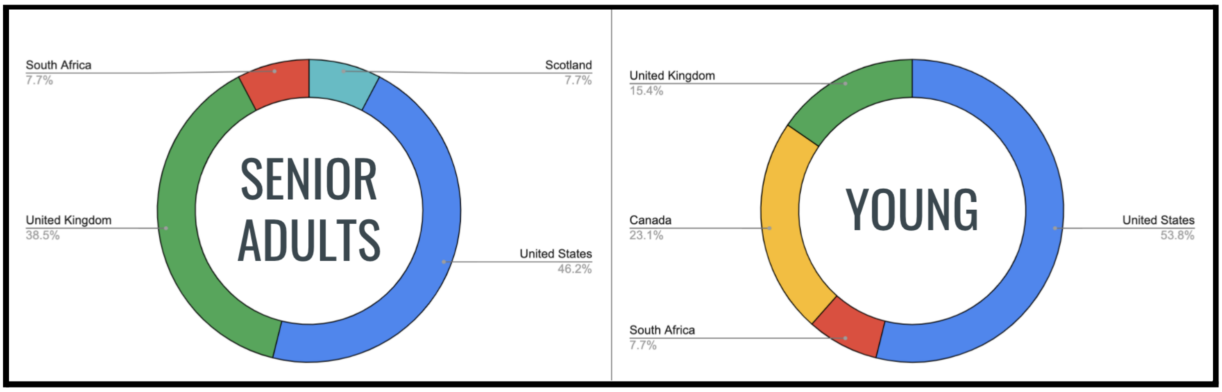


3.3. Participants

We used two groups of participants: **SENIOR** (over 65) and **YOUNG** (between the ages of 18 and 30). The participants were recruited through ads in different Facebook groups. They were all native English speakers, so that language wouldn’t be a confounding variable.

The experimental group (**SENIOR**) consisted of 10 women and 3 men, the youngest was 66 and the oldest was 88 (average age of 71). The control group (**YOUNG**) consisted of 9 women and 4 men, the youngest was 22 and the oldest was 29 (average age of 26). Most of the participants from both groups were Jewish, and currently living in Israel. See Figure 3.6. for country of origin.

Figure 3.6. Country of Origin



3.4. The Experiment

Due to covid-19 restrictions, this experiment was conducted online. I played the role of the wizard, because as the experiment designer, I already knew the domains and the dialogue acts.

During this experiment, the user experienced two different recovery strategies:

- **AskRepeat** - responding with the message: *"Can you please repeat that?"*
- **Reprompt + TerseYouCanSay** - 2-stage strategy:
  - First, responding with a reprompt (repeating the question in a different tone).
  - Following a consecutive error, responding with the message: "Sorry, I didn't catch that. You can Say 'x', 'y' or 'z'", where 'x', 'y' and 'z' are the available options.

The experiment workflow (see Figure 3.7) is comprised of 5 parts:

1. Zoom Briefing:

   A video chat with the participant, in order to see that they're in a quiet environment, and that they're responsive. The briefing included an overview and examining the PDF file sent half an hour before the Zoom meeting. The PDF had 2 pages of general instructions and requirements, followed by 6 cards, visually describing different parameters for table reservation (see Figure 3.8).

2. Pretest:

   The first screen was an informed consent form, where the participant was asked to approve it, in order to start the experiment. They were then presented with general questions by the SDS. The participant was asked about their age, country of origin, etc., to see if their equipment is in working condition. Non-understanding errors were faked in order to see how the participants would respond to the recovery strategies. Strategy A was invoked in a conversation with half of the participants, and Strategy B with the other half.

3. 6 Scenarios:

   Each of the scenarios started with a message “Thank you for calling <restaurant name>. How can I help you?”. According to each of the cards, ordered in the same order for all of the participants, the participant would answer questions and reserve a table. Whether successful or unsuccessful, after the scenario would be over, the participant would see a screen with a button, and would continue to the next scenario whenever they’re ready. Scenarios 1, 3 and 4 invoked the **AskRepeat** strategy, while scenarios 2, 5 and 6 invoked the **Reprompt + TerseYouCanSay** strategy (see Figure 3.8).

4. SASSI Questionnaire:

   After the 6th scenario, the participant was presented with a subset of the SASSI questionnaire (see section 3.5) in a Google form.

5. Zoom Summary:

   In this post experiment conversation, I asked participants about their experience, and asked them to answer questions. At a certain point in the conversation, I exposed the manipulation, and described the WoZ experiment and its goals.

During the experiment, whether in the pretest or while reserving a table, the participant would hear a pre-recorded message accompanied with a visual cue (see Figure 3.5). After the SDS message was over, the visual changed into a bleeping red ball, signaling to the participant that the microphone is recording, and that they can talk now. And this sequence of SDS messages and user responses would continue until an unrecoverable error would occur, or after a successful table reservation. An unrecoverable error was defined as three consecutive non-understanding errors. A transcript of a conversation can be seen in Figure 3.9.

Each scenario consisted of 5 parameters (see Figure 3.10):

1. Date: a calendar with the required day highlighted.
2. Time: a watch indicating the required time.
3. Dietary Restriction: a circle with a symbol indicating what's allowed/forbidden to eat.
4. Party Size: a number with an icon of people, to indicate how many people it is for.
5. Phone number: a number with an icon of a phone, used to complete the reservation.

Figure 3.7. Experiment Workflow

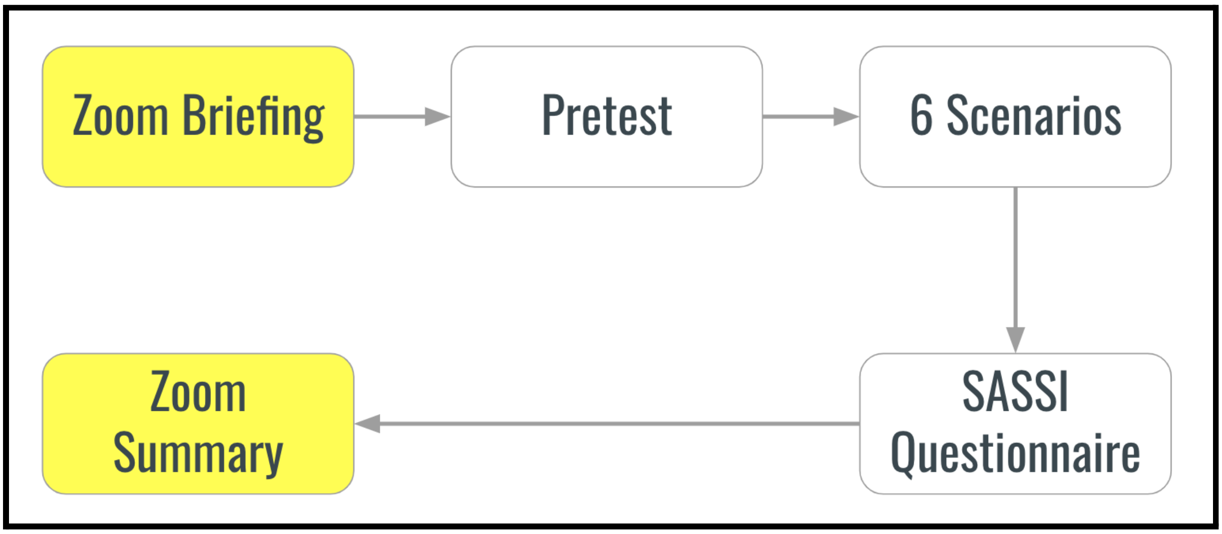
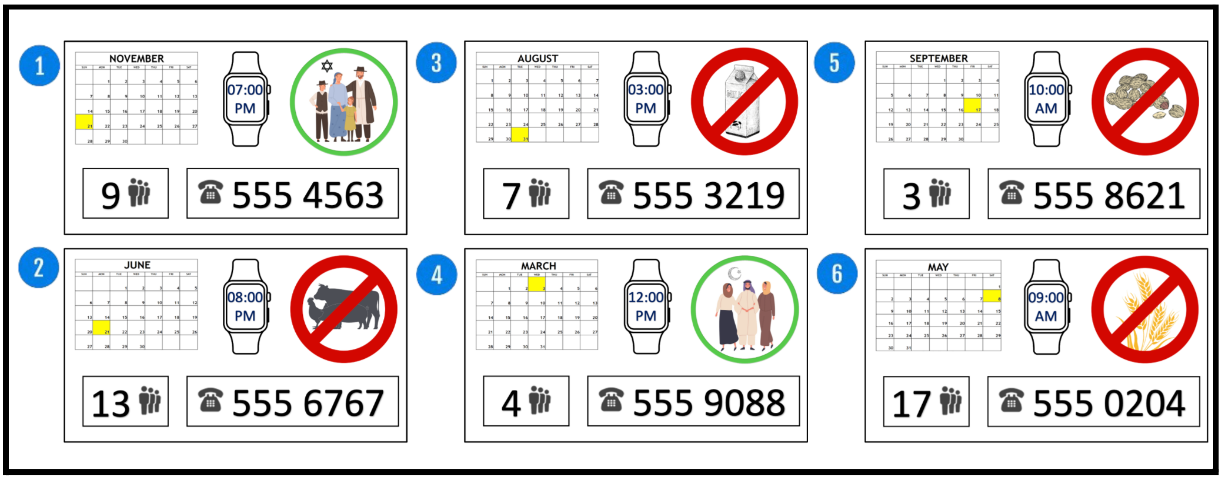


Figure 3.8. Cards Used for Table Reservation



3.5. Metrics for Quality Evaluation

I used the same metrics that Bohus and Rudnicky (2005) used:

1. **Task success**

   This was defined as both binary (success/failure) for each of the 6 scenarios performed by a user.

2. **User satisfaction**

   I used a 1-7 Likert scale to express the user satisfaction, which was elicited through an online post-experiment SASSI questionnaire (Hone and Graham, 2000). The user satisfaction score corresponds therefore to the overall experience the user had with the system. I only used a subset of the SASSI questionnaire - 17 questions grouped in 3 factors:

   - Likability: The system is friendly.
   - Annoyance: The interaction with the system is irritating.
   - Habitability: I always knew what to say to the system.

And in addition:

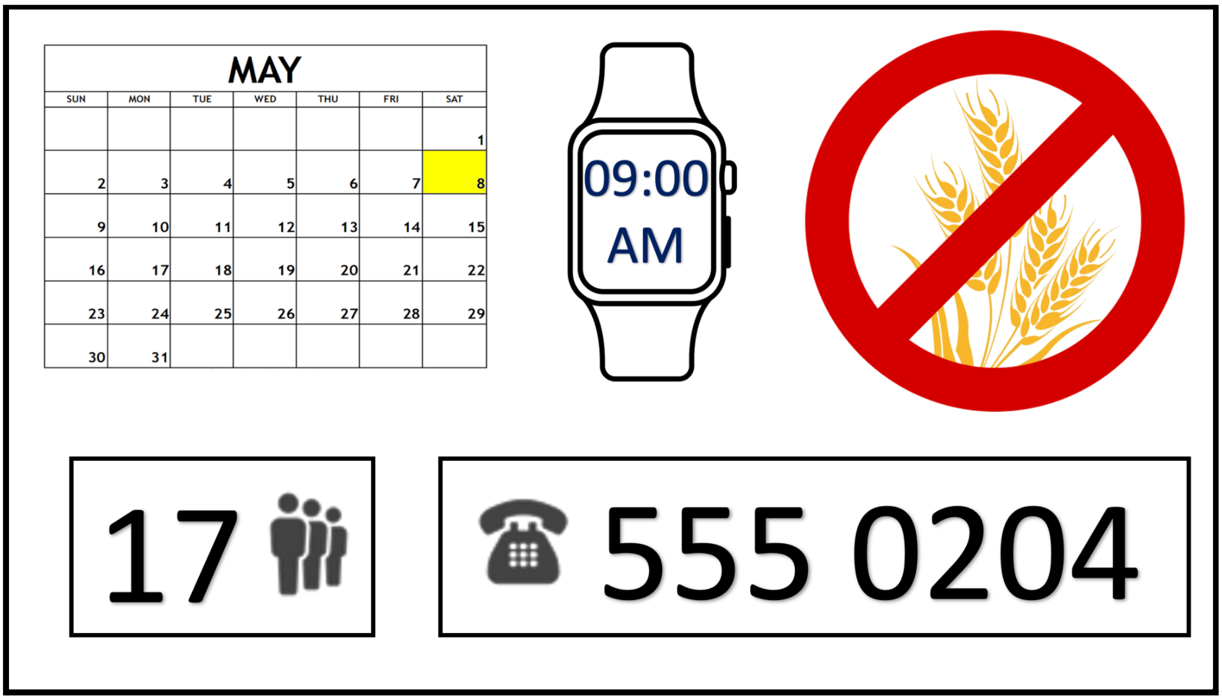
3. **Number of turns taken**

   The average number of turns for each age-group and for each strategy used.

Figure 3.9. Example for a successful conversation with a SENIOR participant (scenario 6)

PYDIAL	Thank you for calling Anastasia. How may I help you?
USER	We would like to make a reservation for May 8 at 9 a.m.
PYDIAL	How many people is it for?
USER	It's a very big party we have 17 it's a very big party we hope you can accommodate us
PYDIAL	How many people is it for?
USER	It's for 17 people
PYDIAL	Are there any dietary restrictions?
USER	Yes there are we cannot have wheat
PYDIAL	Can I take your name and phone number, please?
USER	Sure Christine 555 0204
PYDIAL	We have a table available for Saturday, the 8th of May, at 9AM, the table can sit more than 10 people, and we offer gluten free options.
USER	Wonderful thank you very much we looking forward to it bye
PYDIAL	Thank you. We will see you and your party at 9AM on Saturday, the 8th of May.

Figure 3.10. Card no. 6

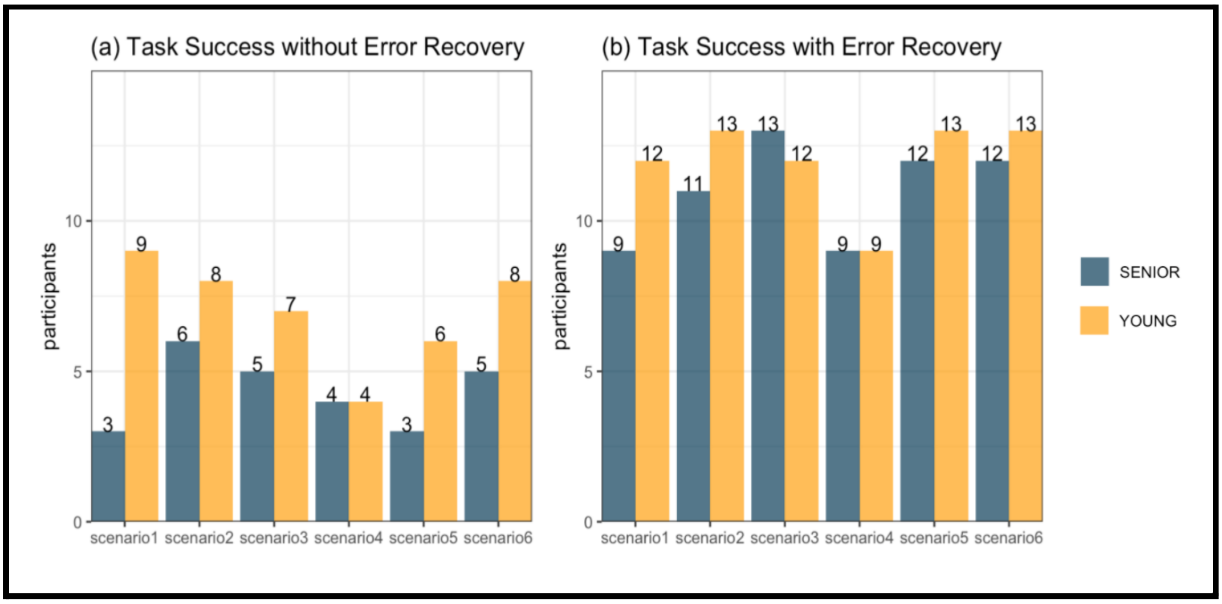


As described in section 3.3, I used 26 native English speaking participants, 13 **SENIOR** and 13 **YOUNG**. In this chapter I'll describe their performance regarding the 6 scenarios of table reservation. I used 3 metrics for quality evaluation: Task success, User satisfaction and Number of turns taken (see section 3.5).

4.1. Task Success

Overall, the SENIOR participants needed more help with recovery than the YOUNG participants. The graph in Figure 4.1. (a) shows the number of participants from each group who didn't have a single non-understanding error. Next to it, (b) shows the total number of participants who managed to reserve a table at each scenario with and without non-understanding errors.

Figure 4.1. Task Success with and without Error Recovery



AskRepeat strategy was used in scenarios 1, 3 and 4, while **Reprompt + TerseYouCanSay** was used in scenarios 2, 5 and 6. Both strategies were found to help participants in completing the task, as can be seen in Figures 4.2. And 4.3. In Figures 4.2. (b) and 4.3. (b), the reader can see the added improvement after the recovery (the percentage from the total number of participants: 13 in each group). Looking at both strategies, the minimal improvement for the

SENIOR participants is 38.5% (scenarios 2 and 4) and the maximal improvement is 69% (scenario 5); The minimal improvement for the **YOUNG** participants is 23% (scenario 1) and the maximal improvement is 54% (scenario 5).

Figure 4.2. AskRepeat Performance

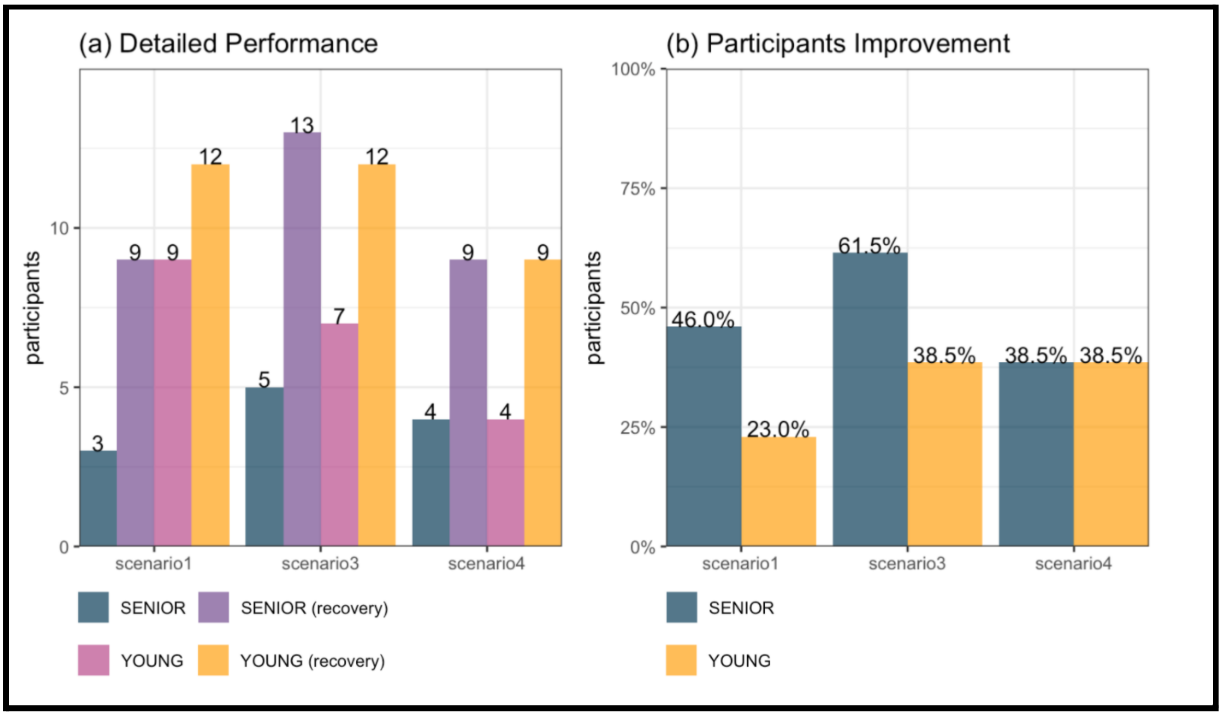
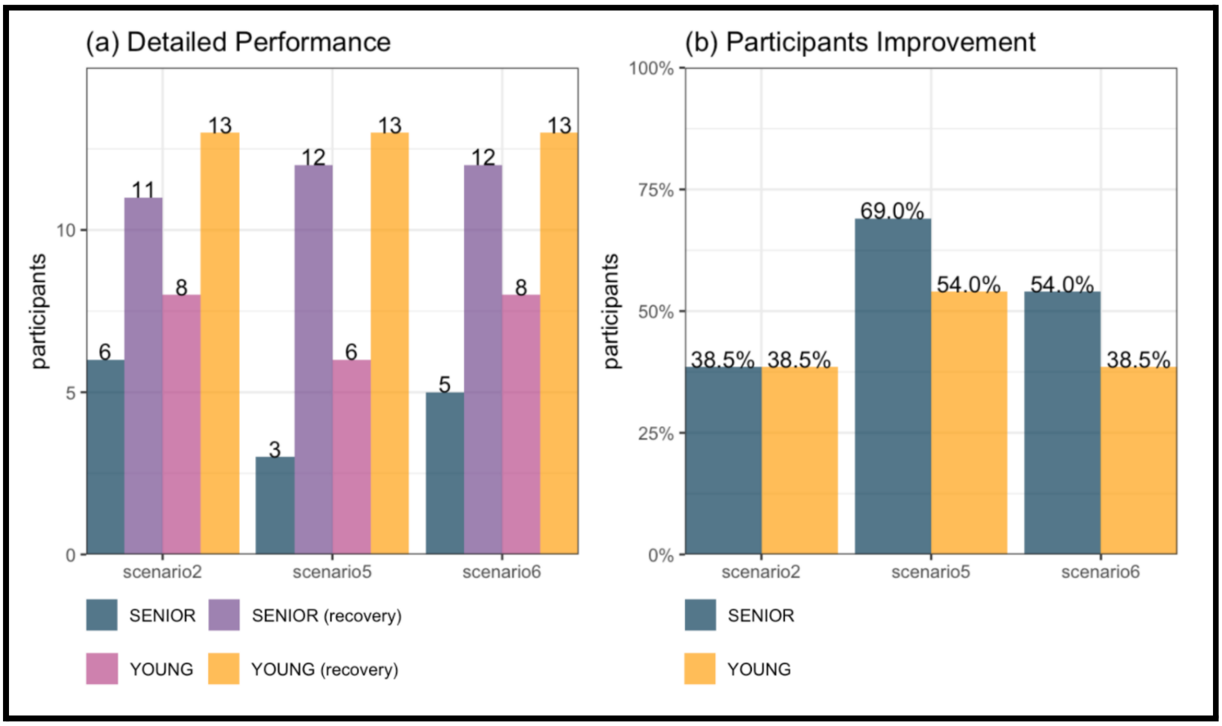
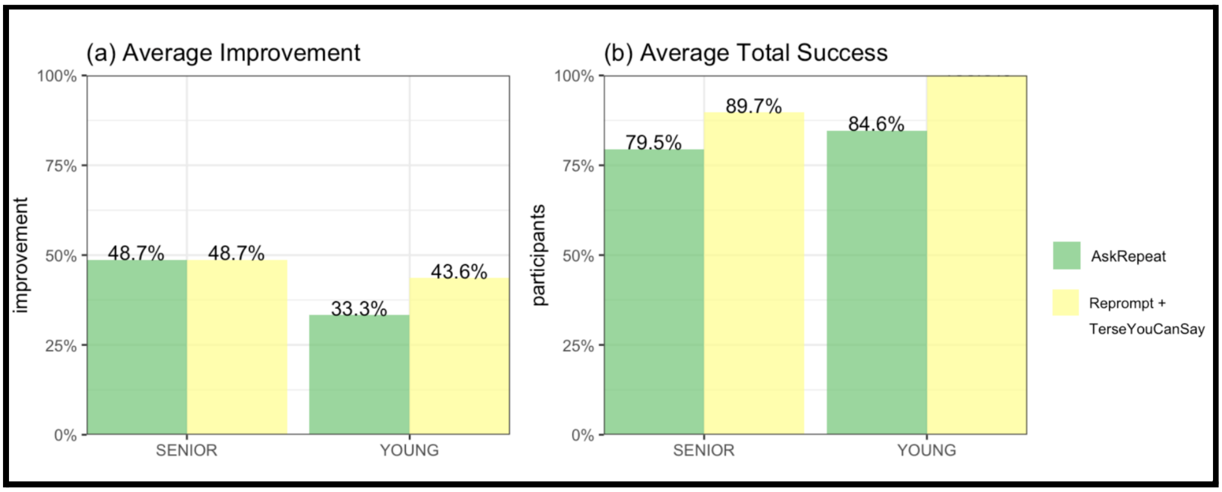


Figure 4.3. Reprompt + TerseYouCanSay Performance



When comparing both strategies (see Figure 4.4), the average improvement is the same for the **SENIOR** participants, but for the **YOUNG** participants, **Reprompt + TerseYouCanSay** is better. When examining the average number of participants successful in reserving a table in the two groups of scenarios, we can see that both **SENIOR** and **YOUNG** participants had a higher success rate in the **Reprompt + TerseYouCanSay** scenarios.

Figure 4.4. Both Strategies Performance



4.2. User satisfaction

In this study, I used a subset of the SASSI questionnaire - 17 questions grouped in 3 factors:

- Likability: The system is friendly.
- Annoyance: The interaction with the system is irritating.
- Habitability: I always knew what to say to the system.

The 1-7 Likert scale's marks are mapped as following:

- 1: Strongly Disagree
- 2: Disagree
- 3: Slightly Disagree
- 4: Neutral
- 5: Slightly Agree
- 6: Agree
- 7: Strongly Agree

4.2.1. Likeability

We can see in Figure 4.5. (a) that the system was more likable by the YOUNG participants. They gave it a score inclining towards "*Slightly Agree*", where the SENIOR participants gave it an average score between "*Slightly Disagree*" and "*Neutral*".

4.2.2. Annoyance

In order to calculate a negative reaction from the statement in the questionnaire, I had to reverse the score for the statement: "*I always knew what to say to the system.*" Meaning I reversed the scale, so 1=7, 2=6, and 3=5.

We can see in Figure 4.5. (b) that the **SENIOR** participants were more annoyed by the system, and gave its annoyance an average score of "*Slightly Agree*", while the **YOUNG** participants gave it an average score of somewhere between "*Neutral*" and "*Slightly Agree*".

4.2.3. Habitability

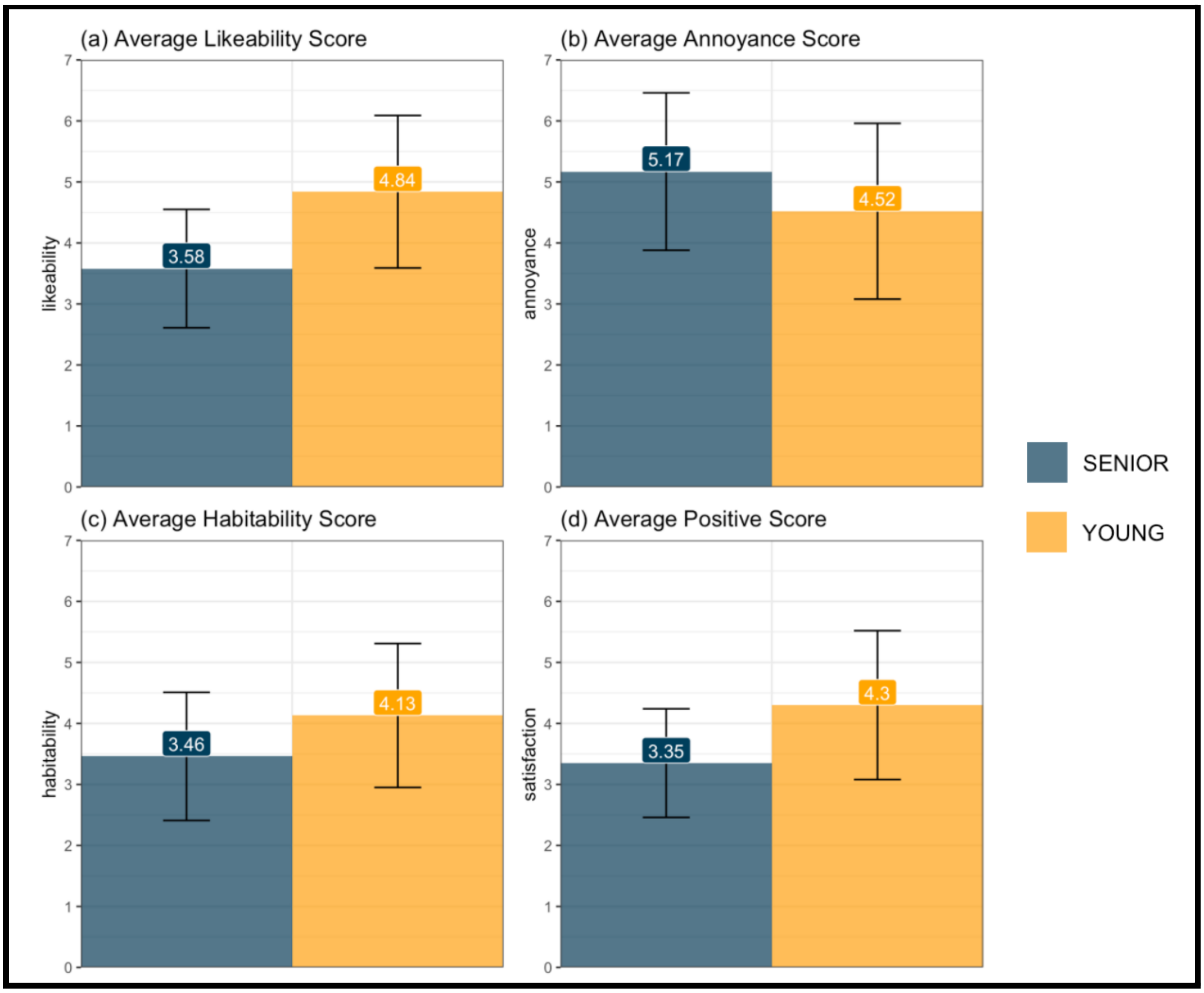
The **SENIOR** participants felt less "at home" than the **YOUNG** participants when using the system, as seen in Figure 4.5. (c). They gave it an average score between "*Slightly Disagree*" and "*Neutral*", while the **YOUNG** participants gave it an average score of "*Neutral*".

4.2.4. User Satisfaction

In order to calculate a total positive score, I had to reverse the scores for 3 "annoyance" statements, which in essence, are negative reactions.

As seen in Figure 4.5. (d), the **YOUNG** participants were more happy with using the system than the **SENIOR** participants. They gave it an average score of "*Neutral*", where the **SENIOR** participants gave it an average score between "*Slightly Disagree*" and "*Neutral*".

Figure 4.5. Average SASSI Questionnaire Scores



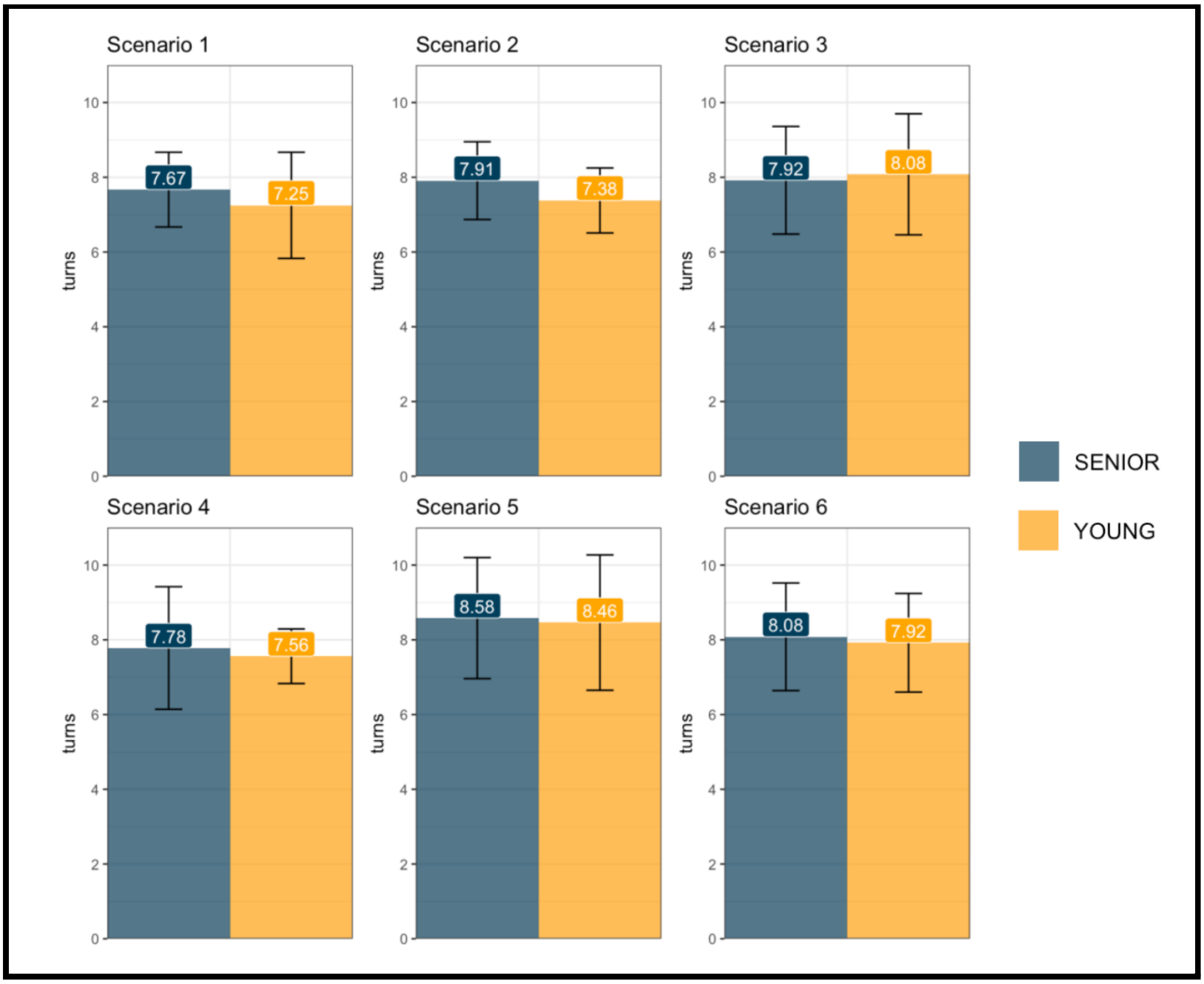
4.3. Number of turns taken

4.3.1. Scenario Analysis

When comparing the performance of the **SENIOR** group to that of the **YOUNG** group, concerning their average number of turns, there isn't a major difference, as evident from Figure 4.6.

It's important to state that only successful scenarios were taken into account.

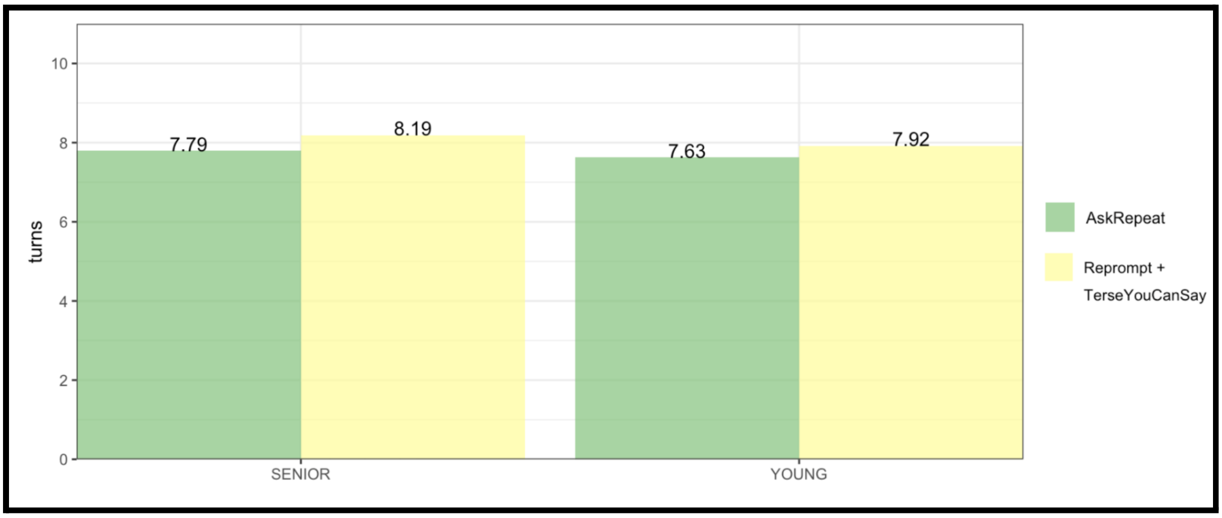
Figure 4.6. Average Number of Turns Taken per Scenario



4.3.2. Recovery Strategy Analysis

When examining the performance per group per recovery strategy, as seen in Figure 4.7, there wasn't a significant difference between the groups or between the strategies.

Figure 4.7. Average Number of Turns Taken per Recovery Strategy



4.4. SENIOR and YOUNG differences

Different Past User Experience

The **YOUNG** participants tend to give one of the reservation parameters in the first message than the **SENIOR** participants. Only 4 of the **SENIOR** participants tried to do that, while 8 of the **YOUNG** participants tried to do that. It can be assumed that the younger participants are more accustomed to using SDS than the older participants, and assumed the SDS could handle the information.

Cognitive Limitations

The **SENIOR** participants experienced some difficulties in performing the task that their **YOUNG** counterparts did not.

- Cognitive Overload

  2 participants (both aged 73) were having difficulties with the TerseCanYouSay strategy concerning dietary restrictions (see the "*restriction*" parameter in Table 3.4): One shouted "*You're talking too fast*", and the other just randomly chose a different restriction without listening.

  Furthermore, they both were confused about the dietary restriction in the first scenario. Roughly a half an hour before the experiment, I sent the participants a PDF containing the scenario cards, and unlike Figure 3.8, each card was on a different page. At the beginning of the file, I added a page with examples for each parameter, and although I emphasised that the experiment has 6 scenarios, which correspond to 6 identical looking cards in the PDF, they still treated the instructions page as data for the first scenario. I think that the Zoom chat before the experiment was too much for them, and couldn't really pay attention, and just wanted to start the experiment.

- Confusion
  - A participant, aged 66, was acting like one of the **YOUNG** participants for the first 5 scenarios, but at the 6th, he was the only one of all the participants in both groups that didn't understand the dietary restriction (see Figure 3.10). The recovery strategy for that scenario was AskRepeat. I think that if he heard

the TerseCanYouSay message describing the different restrictions, he would have completed the scenario successfully. His first answer was “*no cereals*”, and after being asked to repeat, he answered “*no restrictions*”.

- ○ A participant, aged 69, couldn’t understand the positive restrictions in scenarios 1 and 4, both using the AskRepeat recovery strategy, so the options weren’t available to them.
- ○ A participant, aged 74, misread the hour from the card, but after hearing the recovery message, said the correct time.

<u>Vocabulary Differences</u>

8 **SENIOR** participants (61%) used the following phrasing:

- Instead of X AM: “X *in the morning*”.
- Instead of X PM: “*X in the evening*”, “*X in the afternoon*”, “*X o’clock*”.

In comparison, only one **YOUNG** participant (7%), aged 28, used the same phrasing to describe the time for the table reservation scenarios.

One exception was “*12 PM*” which was described as “*noon*” by 3 **YOUNG** participants (23%) and only one **SENIOR** participant (7%). But that is probably because it was scenario 4, and the participant learnt the correct phrasing in previous scenarios.

5. Discussion

5.1. Network Limitations

The experiment had to be conducted online because of the Covid-19 pandemic, and social distancing. This in turn made it possible to recruit participants from all over the world, as the conditions for participation were being a native English speaker at a certain age.

The data communication throughout the experiment was as following (see also Figure 5.1):

1. Automatic Speech Recognition (ASR):
   The participant is somewhere in the world, recording their voice using a microphone. This recording is transferred to an AWS remote machine, located in London, and from there transferred to Google Cloud API (speech-to-text) to be converted into text, sent back to the remote machine.
2. Natural Language Understanding (NLU):
   The text is analyzed by the semantic input component of the specific PyDial server. If the user said anything relevant to the experiment, slots will be filled with values.
3. Dialogue Manager (DM):
   The text from the ASR and the filled slots from NLU will be sent to the terminal's screen of the human wizard in Tel Aviv. The wizard then enters an ID for the next semantic response, transferring the request from his terminal to the remote machine.
4. Natural Language Generation (NLG):
   The semantic response is converted into text in a natural language: requesting data, or informing the user about the filled slots.
5. Text To Speech (TTS):
   A voice recording is sent to the participant's browser and played.

The participant had to wait a few seconds before they heard the spoken response, regardless of the wizard's speed, because of network latency. The latency was caused by the distance between the participant's computer, the wizard's computer and the remote servers. This isn't an ideal situation for a conversation, and some of the participants were upset it was so slow.

5.2. Restarting the Experiment

A small number of the experiments had to be interrupted and restart:

- Network issues: There was a disconnection for some unknown reason.
- Incorrect data: several participants said “2 people” when describing the dietary restrictions, e.g., “2 people from our party are vegetarian”. This changed the data for the party size, and as a wizard, I couldn’t manually do that. The dietary restriction was recorded successfully, and so did all the parameters before that, so I put it as a successful scenario.

Trying to disguise my role as a wizard, I texted the participants and told them that “I see in the logs that there’s a problem and I need to restart the experiment”. They were all very understandable and we continued at the scenario which failed, or the one after that.

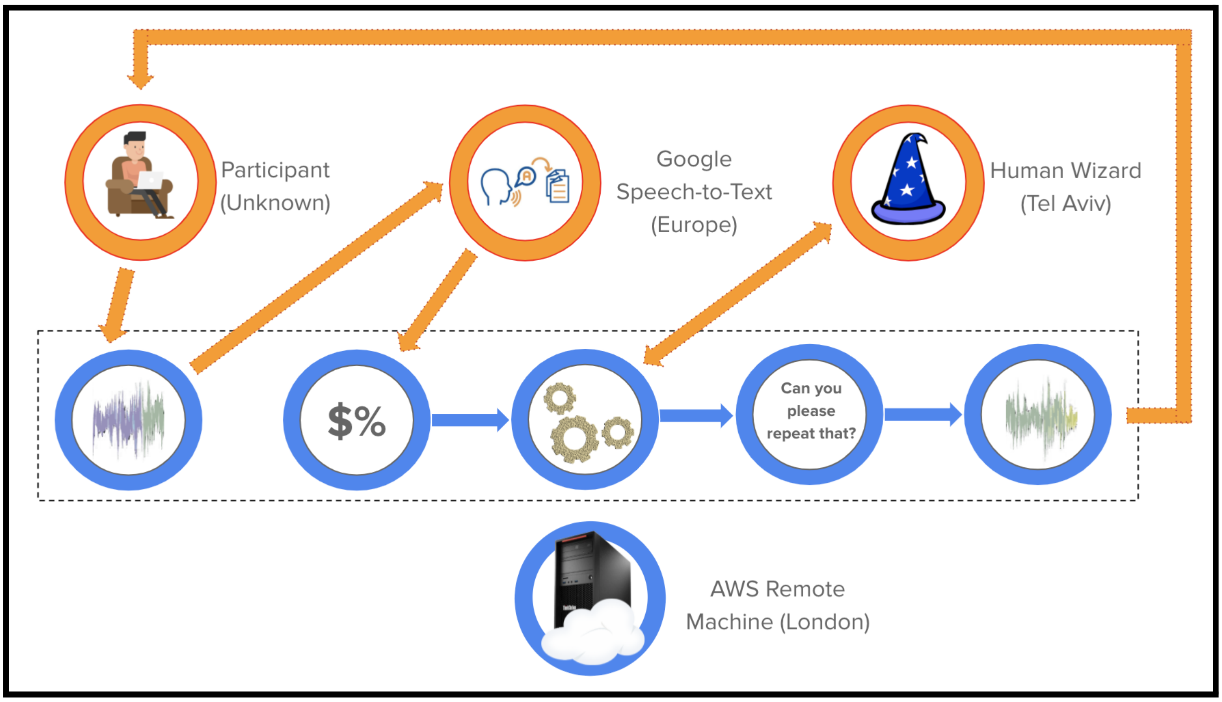
5.3. Voice Recording Limitations

All of the participants used a chrome browser, but they didn’t have the same computer model, of course, or the same operating system. 6 participants were not used for this experiment, because their voices weren’t recorded properly, either because of network problems, or faulty microphones. In addition, for technical reasons, I couldn’t record the participant’s voice until they stopped talking. I gave each recording a fixed timeout of 10 seconds, after which the recording stopped. The participants were warned this could happen, and had a visual cue of a red circle to tell them when to start speaking. Sometimes the participants spoke too early or too late, and their voices weren’t recorded at all, making them angry if they answered correctly.

5.4. Speech Limitations

The preliminary condition to participate in the experiment was to be a native English speaker. The participants are from different countries across the world, and have different accents. Some had difficulties pronouncing foreign words like *Halal* for the dietary restrictions; Some had difficulties pronouncing the numbers *Three* and *Four*. None of the recovery strategies used in this experiment would have helped them with that.

Figure 5.1. A Conversation Turn (communication between computers is in orange arrows)



5.5. Difficult Scenarios

When I designed the different scenarios, I meant to make them as easy and uncomplicated as possible. In addition, I used the help of friends to test out the scenarios, before starting the experiment, and they didn't report any problems. Only when I experimented with diverse groups of participants, that I noticed recurring issues.

<u>AskRepeat: Scenario 4 (March 3rd, 12 PM, 4 people, halal)</u>

This scenario (see Figure 3.8) was difficult for both **YOUNG** and **SENIOR** participants (see Figure 4.2). The **YOUNG** participants had trouble pronouncing the words "*halal*" and "*four*" (see Figure 5.2), and said "*noon*" instead of "*12 PM*" (see section 4.4). The **SENIOR** participants had similar problems, for instance, they also had pronunciation problems with the words "*third*" (March 3rd), "*halal*" and "*four*" (one participant spelled f-o-u-r). In addition, they said "noon" instead of "*12 PM*" and one participant said "*halat*" instead of "*halal*", and another participant didn't understand what the dietary restriction was. Recovery strategies can't help with pronunciation, so I'm not sure if using **Reprompt + TerseYouCanSay** would've helped the participants.

Figure 5.2. Example for a failed conversation with a YOUNG participant (scenario 4)

PYDIAL	Thank you for calling The Elephant House. How may I help you?
USER	Hey I'd love to make a booking please
PYDIAL	For which date?
USER	Wednesday the 3rd of March
PYDIAL	At what time would you like it for?
USER	Will be there around 12 p.m.
PYDIAL	How many people is it for?
USER	It's only going to be for
PYDIAL	Can you please repeat that?
USER	
PYDIAL	Can you please repeat that?
USER	It's for 4 people
PYDIAL	Are there any dietary restrictions?
USER	Yes we are all hello
PYDIAL	Can you please repeat that?
USER	We are hello
PYDIAL	Can you please repeat that?
USER	We are hello
PYDIAL	I'm sorry, I'm having difficulty understanding, would you mind calling back? Thank you in advance! Goodbye.

<u>Reprompt + TerseYouCanSay: Scenario 5 (September 17th, 10 AM, 3 people, peanuts free)</u>

This scenario was also difficult for both **YOUNG** and **SENIOR** participants (see Figure 4.3). When designing this domain, I used keywords I thought the participants would use for each dietary restriction, and I didn't anticipate them to use "*nut allergy*" or "*nuts*", in addition to "*peanuts*" (see Figures 5.3 and 5.4). Reprompt didn't help, because they just repeated the same thing. They only managed to continue and reserve a table after hearing the **TerseYouCanSay** message (see the *restriction* parameter in Table 3.4). I believe that after 2 **AskRepeat** messages, the participants would have kept on saying "*nuts*", which would have led to failure on their behalf, and moving on to the next scenario. There were other issues, which are similar to the other scenarios, and weren't as major.

Figure 5.3. Example for a successful conversation with a SENIOR participant (scenario 5)

PYDIAL	Thank you for calling ManaFood. How may I help you?
USER	I'd like to I'd like to book a table
PYDIAL	For which date?
USER	Friday the 17 of September
PYDIAL	At what time would you like it for?
USER	10 a.m.
PYDIAL	How many people is it for?
USER	3 people
PYDIAL	Are there any dietary restrictions?
USER	One of our party has a nut allergy
PYDIAL	Are there any dietary restrictions?
USER	One person has a nut allergy
PYDIAL	Sorry, I didn't catch that. You can say gluten free, halal, kosher, lactose free, peanut free, vegetarian or no restrictions.
USER	Peanut free
PYDIAL	Can I take your name and phone number, please?
USER	My name is Charles Schulz my phone number is 555 8621
PYDIAL	We have a table available for Friday, the 17th of September, at 10AM, the table can sit up to 5 people, and we offer peanut free options.
USER	Cute
PYDIAL	Thank you. We will see you and your party at 10AM on Friday, the 17th of September.

Figure 5.4. Example for a successful conversation with a YOUNG participant (scenario 5)

PYDIAL	Thank you for calling ManaFood. How may I help you?
USER	I'd like to book a reservation for Friday September 17
PYDIAL	At what time would you like it for?
USER	10 a.m.
PYDIAL	How many people is it for?
USER	Just 3 people
PYDIAL	Are there any dietary restrictions?
USER	One person has a peanut allergy
PYDIAL	Can I take your name and phone number, please?
USER	Lucy down by 5586
PYDIAL	We have a table available for Friday, the 17th of September, at 10AM, the table can sit up to 5 people, and we offer peanut free options.
USER	That sounds perfect
PYDIAL	Thank you. We will see you and your party at 10AM on Friday, the 17th of September.

5.6. Non-understanding Errors: Main Sources

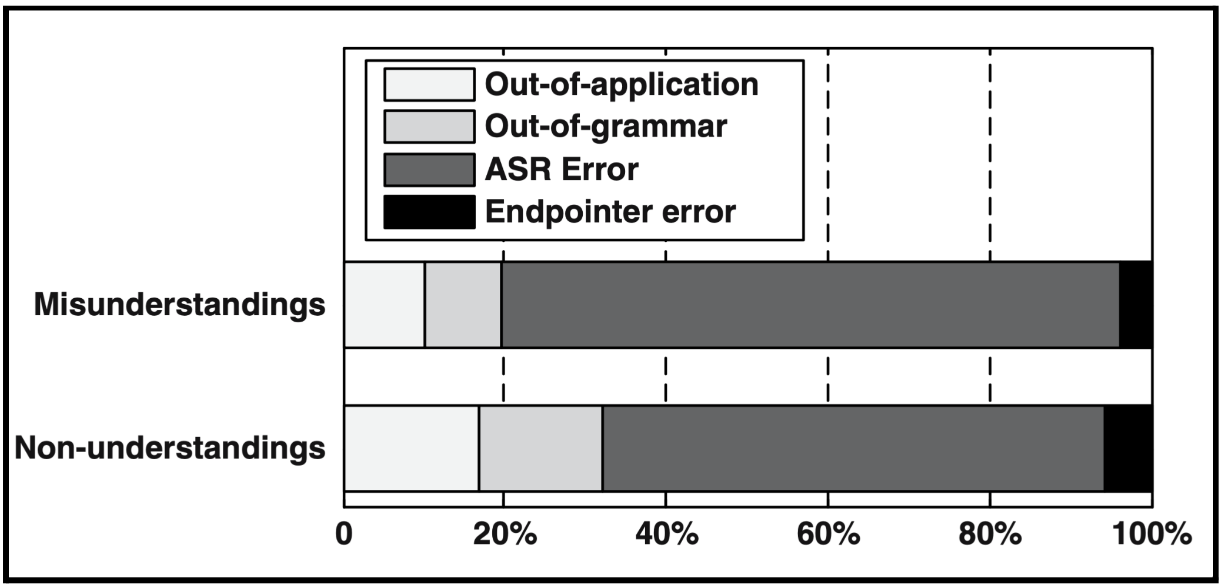
I used almost same sources Bohus and Rudniky (2005) referred to in their analysis (see Figure 5.5):

- <u>Out-of-Grammar Error</u> - The participant used a word the SDS didn’t recognize.
  E.g., “*in the morning*” instead of “*AM*”.
- <u>ASR Error</u> - The participant’s voice wasn’t recognized correctly, generating the wrong text. E.g., “*full*” instead of “*four*”.
- <u>End-pointer Error</u> - There was a problem with the recording of the participant’s voice. E.g., Short or empty utterance.

In addition to the above 3 types, Bohus and Rudniky used “*Out-of-Application*”, where the participant talks about a topic outside of the domain of the application. This is a phenomenon I haven’t encountered in this experiment. It’s either because the participants were all more tech savvy, or because in 2021, people are more used to speaking to SDS than in 2005.

Instead of “*Out-of-Application*”, I added “*Incorrect Data*”, which describes situations where the participant gave different data than what was displayed in the cards. I used 6 pre-recorded messages of successful reservations, so anything that was different than that couldn’t have yielded a successful scenario, and I had to stop the scenario and move on to the next one.

Figure 5.5. Breakdown of Understanding Errors (Bohus and Rudniky ,2005)



As opposed to Bohus and Rudniky's findings, the majority of the errors I found wasn't ASR (see Figure 5.6), the reason could be technological improvements in the 16 years since their experiment. In addition, the source for almost a fifth of the non-understanding errors they found was *Out-of-Grammar*, where in my experiment it was about a half for both the **SENIOR** and the **YOUNG** participants. Another experiment with an SDS using all the recurring phrasings of the participants (see Vocabulary Differences in section 4.4), would've achieved different results. This error type has more to do with the SDS design than with the participants.

Figure 5.6. Breakdown of Error Types

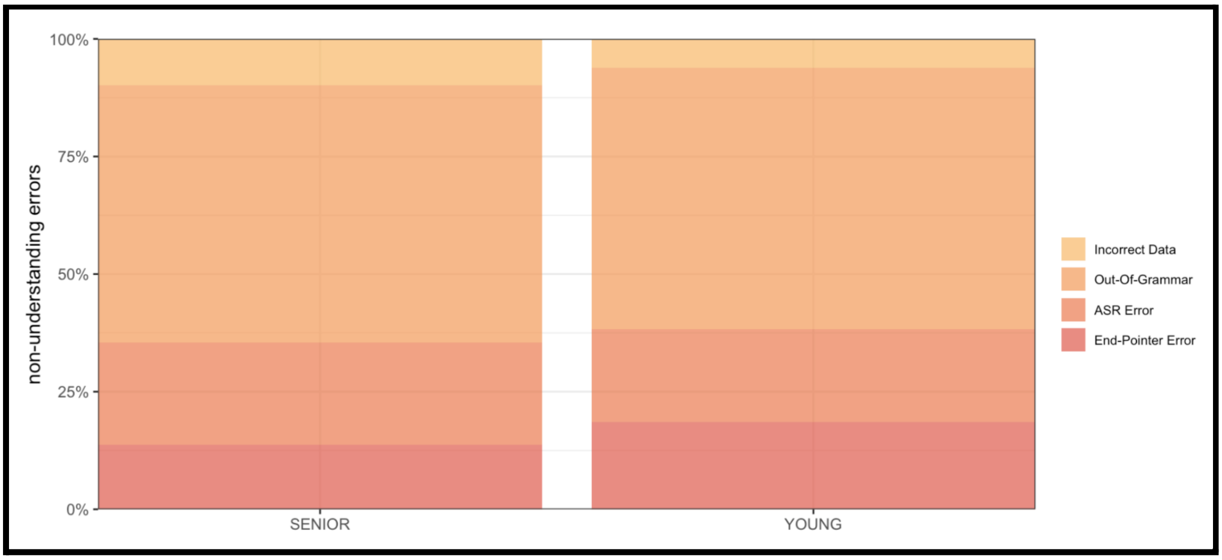
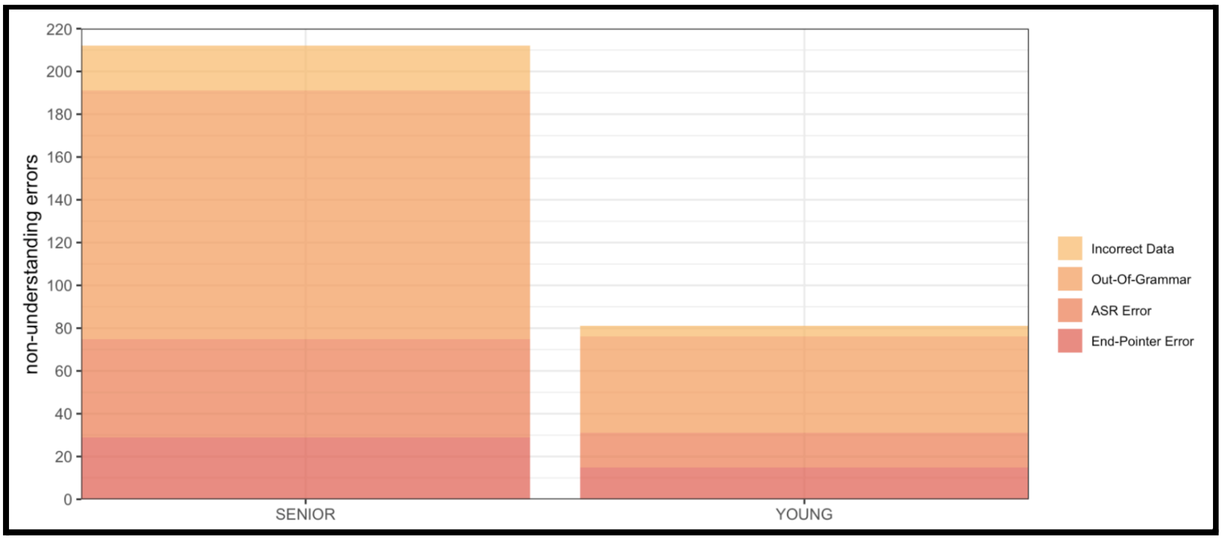


Figure 5.7. Error Types



The **SENIOR** participants had more than twice as many errors than the **YOUNG** participants (see Figure 5.7), although the distribution of the error types was quite similar (see Figure 5.6). The **SENIOR** were more confused than the **YOUNG** participants, and had more "*Incorrect Data*" errors. Another difference is the "*Out-of-Grammar*" errors, where the **SENIOR** participants had 71 more mistakes than the **YOUNG** participants - an increase of 157%. The **YOUNG** participants used a smaller vocabulary and answered straight to the point, where the **SENIOR** participants chose the longer form (*in the morning*) instead of the shorter form they knew (*AM*). The **SENIOR** participants that were chosen for this experiment knew how to get online, use social networks, use a microphone and a camera, and understood technical jargon. It is possible that senior adults who are less tech savvy would have used longer sentences and had even more "*Out-of-Grammar*" Errors.

6. Conclusions

6.1. Success Metrics

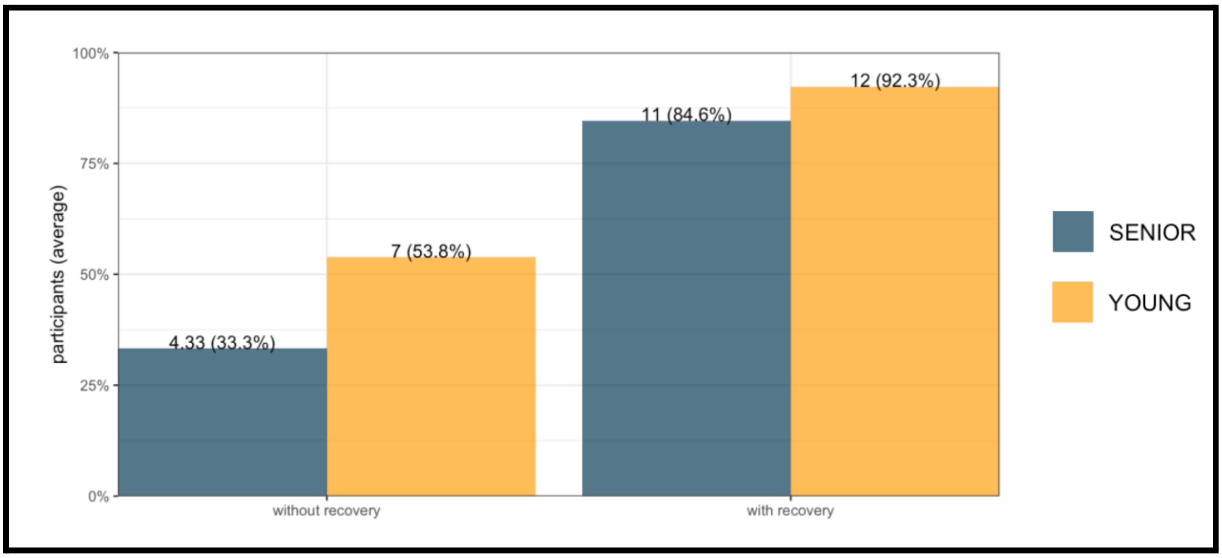
In this study, I used 3 quality evaluation metrics (see section 3.5):

1. Task success.
2. User satisfaction.
3. Number of turns taken.

6.1.1. Task success

In section 4.1, we can see that **SENIOR** participants needed more help recovering from errors than their **YOUNG** counterparts. The average number of **SENIOR** participants to have completed the scenarios successfully without the use of recovery strategies is almost half as much of the average number of **YOUNG** participants (see figure 6.1). After the use of a recovery strategy, the average number of **SENIOR** participants to have completed the scenarios successfully is lower, but almost the same as the average number of **YOUNG** participants.

Figure 6.1. Task Success



When considering the 2 different recovery strategies, we can see in Figure 4.4, that for the **SENIOR** participants, it would seem that there wasn't any difference in recovery, where there was an improvement in recovery for the **YOUNG** participants when using the **Reprompt + TerseYouCanSay** strategy**.**

In section 5.5, I discussed scenario 5, where without using the **Reprompt + TerseYouCanSay** strategy, and without telling the user which word to use to describe the "*peanuts free*" dietary restriction, there wouldn't be any recovery improvement, at all. When looking at Figure 5.6, we can determine that this observation is true, since approximately half of all the non-understanding errors for both groups were "*Out-of-Grammar*" errors.

6.1.2. User satisfaction

In section 4.2.4, we can see that the **SENIOR** participants gave the SDS an overall average score between "Slightly Disagree" and "Neutral", while the **YOUNG** participants gave it an average score of "Neutral". In the post-experiment Zoom chat, the **SENIOR** participants shared their thoughts with me, and said that they would rather talk to a person than a machine, and they were very annoyed by the repetitive nature of the conversation. On the other hand, some of the **YOUNG** participants told me that they would use this SDS if it was available as an app on their phone, and that it was very convenient and straight to the point. It's important to point out to the reader that although the **SENIOR** participants weren't very satisfied, in a real case scenario, I believe that they would rather use an SDS than hear a series of busy signals when calling a restaurant, and wait for the line to be free, in order to complete a simple task as reserving a table in a restaurant. Of course, this is only a speculation that could be confirmed or refuted experimentally, and only when the alternative isn't just calling a different restaurant.

6.1.3. Number of turns taken

In section 4.3.2, I discussed the average number of turns taken. I examined the average number of turns between the age groups and between the different scenarios. In general, the YOUNG participants needed less turns to complete the scenarios successfully (see Figure 4.7), but the difference wasn't a significant one.

6.2. Optimal Recovery Strategy

Is one of these recovery strategies better for older people?

We saw in section 6.1.1, that older people need more help completing a task than younger people. In addition, in section 5.6, we also saw that they had a higher number of non-understanding errors, specifically "*Out-of-Grammar*" errors.

When using a vocabulary limited SDS, it would be hard to try and guess the right word the SDS is hoping the user will use. As we saw in the analysis of scenario 5 in section 5.5, the only strategy that could have helped was the **Reprompt + TerseYouCanSay** strategy. In other scenarios or when using SDS with an unlimited vocabulary, it is possible that any strategy would have been sufficient.

Is the optimal recovery strategy different for younger and older people?

When analysing the sources of non-understanding errors of the different participants, we saw that error type distribution was very similar for the **SENIOR** and the **YOUNG** participants, although the **SENIOR** participants had more than twice as many errors (see Figures 5.3 and 5.4).

In Figure 4.1, we saw that for scenario 5, the **YOUNG** and **SENIOR** had the same problems and same improvement when using **Reprompt + TerseYouCanSay** strategy. It would seem, that as with the **SENIOR** participants, the **YOUNG** participants need to hear a list of options, in order to understand which is the correct phrasing for their required parameter.

6.3. Further Research

The participants were very annoyed with the waiting times between each system response. In section 5.1, I described the network limitations, so it would be interesting to see a similar experiment being done with an offline fully automated SDS, without a human wizard. In addition, it would be better to use populations with a neutral accent, so that scenarios wouldn't fail because of inadequate acoustic models. Another issue that would be interesting to explore is using more scenarios with phrases which are less immediate to the participants, and see how they react with both strategies. I would suggest to determine term immediacy by a frequency count, using a large corpus relevant to the tested population. Lastly, I want to address the size of the tested population. In this study I used 26 participants, 13 of each age group. Perhaps an experiment with larger populations would reveal further generational differences or understanding errors.

References

Bohus, D. and Rudnicky, A.I. 2005. Sorry, I didn't catch that!-An investigation of non-understanding errors and recovery strategies. In *6th SIGdial workshop on discourse and dialogue*.

Brookmeyer, R., Johnson, E., Ziegler-Graham, K. and Arrighi, H.M. 2007. Forecasting the global burden of Alzheimer's disease. *Alzheimer's & dementia*, *3*(3), pp.186-191.

Dahlbäck, N., Jönsson, A. and Ahrenberg, L., 1993. Wizard of Oz studies—why and how. *Knowledge-based systems*, 6(4), pp.258-266.

Henderson, M., Matheson, C. and Oberlander, J., 2012. Recovering from non-understanding errors in a conversational dialogue system. In *Proceedings of SemDial 2012 (SeineDial): The 16th Workshop on the Semantics and Pragmatics of Dialogue* (p. 128).

Hone, K.S. and Graham, R., 2000. Towards a tool for the subjective assessment of speech system interfaces (SASSI). *Natural Language Engineering*, 6(3-4), pp.287-303.

Kim, J., Jeong, M. and Lee, S.C., 2019, September. " Why did this voice agent not understand me?" error recovery strategy for in-vehicle voice user interface. In *Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications: Adjunct Proceedings* (pp. 146-150).

Kobayashi, M., Kosugi, A., Takagi, H., Nemoto, M., Nemoto, K., Arai, T. and Yamada, Y., 2019, September. Effects of age-related cognitive decline on elderly user interactions with voice-based dialogue systems. In *IFIP Conference on Human-Computer Interaction* (pp. 53-74). Springer, Cham.

Kopp, S., Brandt, M., Buschmeier, H., Cyra, K., Freigang, F., Krämer, N., Kummert, F., Opfermann, C., Pitsch, K., Schillingmann, L. and Straßmann, C., 2018. Conversational assistants for elderly users–the importance of socially cooperative dialogue. In *Proceedings of the AAMAS Workshop on Intelligent Conversation Agents in Home and Geriatric Care Applications co-located with the Federated AI Meeting* (Vol. 2338).

Koulouri, T. and Lauria, S., 2009. A WOz framework for exploring miscommunication in HRI. In *Procs. of the AISB Symposium on New Frontiers in Human-Robot Interaction* (pp. 1-8).

Mihailidis, A., Boger, J.N., Craig, T. and Hoey, J., 2008. The COACH prompting system to assist older adults with dementia through handwashing: An efficacy study. *BMC geriatrics*, 8(1), p.28.

Olaso Fernández, J.M., 2017. Spoken dialogue systems: architectures and applications.

Opfermann, C. and Pitsch, K. 2017. Reprompts as error handling strategy in human-agent-dialog? User responses to a system's display of non-understanding. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (pp. 310-316). IEEE.

Roth, T.N., Hanebuth, D. and Probst, R. 2011. Prevalence of age-related hearing loss in Europe: a review. *European Archives of Oto-Rhino-Laryngology*, *268*(8), pp.1101-1107.

Skantze, G. 2003. Exploring human error handling strategies: Implications for spoken dialogue systems. In *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*.

Ultes, S., Barahona, L.M.R., Su, P.H., Vandyke, D., Kim, D., Casanueva, I., Budzianowski, P., Mrkšić, N., Wen, T.H., Gasic, M. and Young, S., 2017, July. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations* (pp. 73-78).

Vipperla, R., Renals, S. and Frankel, J. 2008. Longitudinal study of ASR performance on ageing voices. In *Proc. Interspeech, Brisbane, 2008*, pp. 2550–2553.

Wolters, M., Georgila, K., Moore, J.D. and MacPherson, S.E., 2009a. Being old doesn't mean acting old: How older users interact with spoken dialog systems. *ACM Transactions on Accessible Computing (TACCESS)*, *2*(1), pp.1-39.

Wolters, M., Georgila, K., Moore, J.D., Logie, R.H., MacPherson, S.E. and Watson, M., 2009b. Reducing working memory load in spoken dialogue systems. *Interacting with Computers*, *21*(4), pp.276-287.

תוכן עניינים

תקציר

סייענים קוליים, כדוגמת סירי של אפל, נמצאים בשימוש בכל מקום, למרות שהמשתמשים עדיין עשויים להיות לא מובנים על ידי התוכנה. כאשר יתקל בשגיאה שכזו, הסייען ינסה "להציל" את השיחה, בעזרת שימוש באסטרטגיית התאוששות. יתכן שקשישים יאלצו להתמודד עם בעיות אלו בתכיפות רבה יותר מאנשים צעירים, מכיוון שהם אינם "ילידים דיגיטליים", והציפיות שלהם משיחה עם מערכת ממוחשבת עשויות להיות לא מותאמות. כמובן, דבר זה ישתנה עם הזמן, ויותר אנשים יהיו "ילידים דיגיטליים" מגיל צעיר.

בעבודה זו, אתחיל בסקירת מחקרים קודמים בתחום של אינטרקציה בין אדם למכונה ואסטרטגיות התאוששות. במחקרים אלה, התעלמו מאוכלוסיית הקשישים כליל, או שהתמקדו בקשישים אך לא באסטרטגיות ההתאוששות. מטרת מחקר זה היא לגשר על הפער הזה, וללמוד איזו אסטרטגיית התאוששות מועדפת על בני ובנות הגיל השלישי, והאם אסטרטגיית ההתאוששות האופטימלית שנמצאה עבור קשישים וקשישות שונה מזו שנמצאה עבור צעירים וצעירות.

שחזרתי את ניסוי "הזמנת המקום" מסוג הקוסם-מארץ-עוץ של Bohus and Rudnicky, והחלפתי את חדרי הישיבות בשולחנות במסעדה. נעזרתי בנסיינים ונסייניות משתי קבוצות גיל שונות של דוברי אנגלית כשפת אם, האחת עם 13 אנשים בין הגילאים 18 עד 30, והשנייה עם 13 אנשים מעל גיל 65. הם התבקשו לשוחח עם מערכת הזמנות מקוונת באמצעות אתר אינטרנט, כאשר במהלך השיחה, 2 אסטרטגיות התאוששות שונות הופעלו בכל פעם שהתרחשה שגיאה. במהלך הניסוי, הם ביצעו 6 תרחישים שונים של הזמנת שולחן, לפי פרמטרים קבועים מראש, בעזרת אוצר המילים שלהם. התוכנה פותחה ספציפית עבור מחקר זה, באמצעות PyDial, תשתית משוכללת ליצירת מערכות דיאלוג.

במחקר זה, בחנתי כיצד הנסיינים הגיבו לאסטרטגיות התאוששות שונות, תוך שימת דגש על הבדלים בין קבוצות הגיל. גיליתי שבתרחיש 5, כאשר המערכת ציפתה ל-"*peanuts*", מילה פחות מיידית עבור הנסיינים מהמילה "*nuts*", רק אחת מהאסטרטגיות הצליחה לגרום להם להתנסח מחדש ולומר "*peanuts*", לסייע להם להתאושש מהשגיאה ולהמשיך בהזמנה.

אוניברסיטת תל אביב
הפקולטה למדעי הרוח ע״ש לסטר וסאלי אנטין
בית הספר לפילוסופיה, בלשנות ולימודי מדע
החוג לבלשנות

אסטרטגיות התאוששות בדיאלוג
בין קשישים לסוכן חכם

חיבור זה הוגש כעבודת גמר לקראת התואר
״מוסמך אוניברסיטה״ - .M.A באוניברסיטת ת״א
על-ידי
שאול אשכנזי

העבודה הוכנה בהדרכת

פרופ׳ מירה אריאל
אוניברסיטת תל אביב

פרופ׳ בוני וובר
אוניברסיטת אדינבורו

דר׳ מריה וולטרס
אוניברסיטת אדינבורו

אוקטובר 2021