KTH Computer Science
and Communication

Usability of Text and Speech as Input Methods for Natural Language Interfaces

ANGELINA VON GEGERFELT
KASHMIR KLINGESTEDT

Degree Project in Computer Science, DD143X
Supervisor: Arvind Kumar
Examiner: Örjan Ekeberg

Stockholm, Sweden April 18, 2016

Abstract

Today an increasing amount of systems make use of Natural Language Interfaces (NLIs), which make them easy and efficient to use. The purpose of this research was to gain an increased understanding of the usability of different input methods for NLIs. This was done by implementing two versions of a text-based game with an NLI, where one version used speech as input method and the other used text. Tests were then performed with 9 test users that all played through both versions of the game and then evaluated them individually using the System Usability Scale. It was found that text was better as input method on all aspects. Speech however scored high when the users felt confident in their English competence, acknowledging the possibility of using speech as input method for NLIs.

Sammanfattning

Användbarhet av Text och Tal som Inmatningsmetoder för Naturligt Språkgränssnitt

Idag använder en ökande mängd system naturliga språkgränssnitt, vilket gör dem enkla och effektiva att använda. Syftet med denna forskning var att få en ökad förståelse för användbarheten av olika inmatningsmetoder för naturliga språkgränssnitt. Detta gjordes genom att skapa två versioner av ett textbaserat spel med ett naturligt språkgränssnitt, där en version använde tal som inmatningsmetod och andra använde text. Tester utfördes sedan med 9 testanvändare som alla spelade igenom båda versionerna av spelet och sedan utvärderade dem individuellt med hjälp av System Usability Scale, ett system för att mäta graden av användbarhet. Det konstaterades att text fungerade bättre som inmatningsmetod ur alla aspekter. Tal fick dock en hög poäng när användarna kände sig säkra på sin engelska kompetens, vilket talar för möjligheten att använda tal som en inmatningsmetod för naturliga gränssnitt.

Contents

Chapter 1

Introduction

A system that has a Natural Language Interface (NLI) enables the user to interact with the system using natural language, which is a language that has developed as a method of communicating between people (Cambridge Dictionaries Online, 2016). The input method may vary, where some examples are speech, text and body language. A system like this is practical since the user does not have to learn a new language in order to use the system effectively, such as a programming language or specific button sequences.

Development of natural language as an input method for interacting with systems has been an ongoing process since the late 1940s. Back then the work was focused on machine translation with goals such as translating text or speech from one language to another (Jones, 2001). Today implementations of NLIs are highly encouraged and many frequently used systems include it, such as Google Search and Apple's Siri. Many systems with NLIs are open source, meaning anyone can make use of and contribute to the development of even better NLIs.

The research presented in this paper aims to evaluate and compare two different natural language input methods, namely text and speech. This is done by the use of gamification, where the game is inspired by existing text-based adventure games. The evaluation is based on the ISO-definition of usability, which focuses on the effectiveness, efficiency and satisfaction of a product. (ISO.org, 1998)

1.1 Problem Statement

Comparing the two natural language input methods text and speech, which one has the highest usability level according to the ISO-definition of usability?

1.2 Scope

The area of natural languages include additional types of communication other than text and speech, such as body language and touch, although in this research the focus is solely on text and speech. There are also different kinds of NLIs for both of these natural languages. For example one text-based NLI could use strictly grammatical processing while another could be less strict in this matter. In this research a less grammatically strict NLI is implemented for both text and speech, since the main goal is to compare the usability of these two input methods which makes the grammatical strictness less relevant.

1.3 Purpose

This research has been done in order to gain an increased understanding of the usability of text and speech input when it comes to Natural Language Interfaces. This knowledge may then be applied when constructing similar NLIs in the future or to improve the quality of part-of-speech taggers and speech recognition on the whole.

1.4 Terminology

Expression	Abbrevation	Deffinition
Natural Language Interface	NLI	A way for the user to interact with a system or program by the use of human natural language
Natural Language Processing	NLP	Derives meaning from natural language input and converts it into something the computer can understand and vice versa
System Usability Scale	SUS	System to measure level of usability

Table 1.1. Short definitions of relevant expressions

Chapter 2

Background

2.1 Natural Language Interface

A Natural Language Interface (NLI) is a way for the user to interact with a system or program in a more natural and intuitive way. Systems using NLIs have many advantages over other systems when it comes to effectiveness, required pre-knowledge and flexibility. An NLI's primary ability is that it is implemented taking in consideration the user's view of the system and translates this into something the system can understand and execute (Hendrix, 1982). A few examples of frequently used systems using NLIs are Google Search, Wolfram Alpha, Siri and different navigation systems.

2.1.1 Natural Language Processing

Natural Language Processing (NLP) explores how computers can be used to understand and manipulate natural language (Chowdhury, 2003). The input may consist of text, speech or other. NLP can be used for translation into another language, to comprehend and represent the contents of the input, to build or search a database and to maintain a dialogue with a user as part of an interface for database/information retrieval (Allen, 2003). NLP is a necessary part of the back-end of any NLI.

2.2 Speech Recognition

Systems using speech recognition (SR) have been researched and developed as a worldwide activity because of the potential this brings for applications, such as to have voice-interactive management, voice dictation and spoken language translation. Although there have been many successes in the development of practical and useful SR systems, there are still limitations to what can be done. The speech signal is one of the most complex signals that humans are dealing with. In addition to this, there is also the fact that the human vocal system differ between individuals, and phrases can be expressed or pronounced in different ways. However, various successful SR systems have nonetheless been integrated into consumer-technology, such as Google Now. (Lee et al., 1996, page 2)

2.3 Text-Based Adventure Games

Text-based games are a form of interactive fiction, which was the first step away from media where the player is only an observer, such as movies and books, to a media in which the player plays a part of the world. In text-based games the player input commands to change the state of the game. The command's form ranges from verb-noun pairs (such as "go west") to complex sentences with multiple commands ("open the door with key and then go west").(Sweetser, 2008, page 54-55)

2.3.1 Zork

One of the earliest text-based games was "Zork, The Great Underground Empire", which was released in 1980 to critical acclaim. Byte Magazine said that the game was "[...] entertaining, eloquent, witty and precisely written" (Liddil, 1981, page 264). Zork's biggest selling point was its ability to accept free-form instructions. Commands could be put in the same sentence and it would still work, for example "eat the lunch and drink the water" which would consume both items while satisfying hunger and thirst. This created a level of freedom for the player while still being able to accept more complex input.(Liddil, 1981)

2.4 Usability

Usability is defined by the ISO-definition as the extent to which a system or product can be used by users in order to achieve specified goals. The focus is on effectiveness, efficiency and satisfaction in a specific context of use (ISO.org, 1998). In 1986 John Brooks created a way to test a user interface and its usability where he based it on the first edition of the ISO-definition. The idea is that a user tries out a system after which they answer a specific questionnaire concerning the usability of the system. The user should not think for a long period of time or discuss their opinion with anyone before or while filling out the questionnaire. It is important that the user's initial thought and own experience is recorded. The questionnaire consists of 10 statements and the user must rank each statement by a scale of 1-5, where 1 is "strongly disagree" and 5 is "strongly agree". Some examples of the statements are "I thought the system was easy to use" and "I thought there was too much inconsistency in this system", see appendix A.1 figure A.1 for all the questions. (Brooke, 1996)

2.5 Previous Research

2.5.1 NLI Technology in Computer Games

A degree-project in Computer Engineering from a technical university in Spain aimed to see if they could improve the usability of text-based games by using a Natural Language Interface. In it they have three games, one where input is strict commands with no freedom on what the user can input. The other two uses different types of NLI, one without lexical consistency (just a verb and a noun is needed) while the other requires lexical consistency (i.e., forcing the player to use complete and functioning sentences). They found that the system without lexical consistency had the highest usability closely followed by the one with lexical consistency. The strict commands-version were rated very low showing that NLI improves usability of a system. (Ribes, 2015)

2.5.2 Degree Project in Computer Science

A degree project in computer science carried out at KTH in 2015 researched whether speech recognition is a useful input method for natural language. This was done by first developing a game based on the text-based game Zork, where the user would enter commands in natural language. The game was then used to conduct a usability test which included two groups of test users, where the first group used keyboard input and the second group used speech recognition. Data regarding the user's efficiency, behaviour and perception of the system was gathered. The results showed that speech recognition has lower efficiency based on time and number of errors. The users did however perceive speech recognition more positively. The conclusion was drawn that speech recognition can be considered a viable input method in cases where efficiency is not of crucial significance. (Larsson and Qvarfordt, 2015)

Chapter 3

Method

3.1 The Game

The game is inspired by the previously mentioned Zork. It consists of a few rooms and tasks to be performed before reaching a victory scenario. The game differs in environments and plot between the different implementations of NLIs, which requires the user to input different commands in order to win. One version uses typing to control your character's actions and the other uses speech. The reason we decided to create two implementations is so that a user who has played one control-scheme could still play the other without having the benefit of knowing what is required to win.

3.1.1 Plot

The user play as a bunny that has escaped its cage and is on the hunt for food. They need to eat three crackers in each game in order to ease their hunger and win the game. In the speech version the user is a house-pet and is in an apartment and has the ability to be in the kitchen, the living room and the bedroom. In the text version they are a class-pet in a school and can visit the classroom, the hallway and the cafeteria.

3.2 Implementation

When implementing the different versions of the game existing open source libraries were used for word tagging and speech recognition, which were then linked to our own built parser. The parser takes two words as arguments: one verb and one noun. These words are sorted out from the user's command line using the word tagger. The parser then generates the proper response by first handling the verb and then linking the action to the given noun. Verbs handled in the parser are "go", "look", "take", "eat" and "use". Several synonyms to these verbs are also handled by first sending them through a synonym checker that converts them to one of the five verbs handled by the parser. If the verb is not recognized the game responds with "Try something else".

3.2.1 Programming Language

The choice of programming language depended on which existing libraries to be used in the game. Java was convenient to use since both of us were comfortable using it and there were many libraries to choose from that were adjusted to work in Java.

The development of the game was done in an integrated development environment (IDE) called Eclipse, which provide a lot of useful tools whereof some for easily handling dependencies for all the various libraries used. To synchronize our coding progress with each other Git was used, which is a popular version control system.

3.2.2 Stanford POSTagger

The library which was used for tagging command words is the Stanford Part-Of-Speech Tagger. It is part of the Stanford CoreNLP, which is a suite of core NLP tools. A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word, such as noun, verb, adjective, etc. (The Stanford NLP Groups, 2015)

By tagging each word in the user's command line it was possible to sort out which command words were verbs and nouns, the types handled in the parser. By doing this the user was able to communicate with the game using natural language. For example commands like "take the toy located under the couch in the kitchen" will be handled as "take toy", since those words are the verb and noun in the command. Some commands run through the Stanford POSTagger will however be tagged incorrectly if proper grammar is not used. For example when using the command "use key on door" the word "use" gets tagged as a noun instead of a verb, however if you instead use the command "use the key on the door" the word "use" gets tagged as a verb. To handle this issue all nouns were run through the synonym checker as well to see if it matched any of our verbs.

3.2.3 Sphinx4

The Sphinx4 speech recognition system is the latest addition to Carnegie Mellon University's repository of the Sphinx speech recognition systems. It is universal in its acceptance of various kinds of grammars and language models, types of acoustic models and feature streams. Sphinx 4 is developed entirely in the Java programming language and is widely used, which made it suitable for use in the game. (CMU Sphinx, 2015)

Sphinx4 is used solely in the speech version of the game, where the user give commands through speech using a microphone. When a command is spoken, Sphinx4 recognizes the separate words and then converts the command to text. It is then sent to the Stanford POSTagger, etc.

Which words and command structures Sphinx4 can recognize is specified in grammar files (with extension .gram). For example specific verbs and nouns can be specified and then the command structure can be set as *<verb>* *<noun>*, which would make Sphinx4 recognize commands like "use key" but not commands like "use the small golden key". In the game the speech command structure is set as

<command> = <verb> <conjunction> <determiner> <noun> | <cmd>.

The straight line symbolizes "or", so either structure separated by the straight line is acceptable. The conjunctions and determiners are optional, making both commands like "go to the kitchen" and "go kitchen" recognizable. The *<cmd>* contain special commands like "quit" and "help".

When implementing Sphinx4 into the game it turned out that the more recognizable words, the higher risk of Sphinx4 misinterpreting the spoken command. Although, cutting down on the amount of synonyms would make the input less natural language like. We found a balance between amount of synonyms and recognition by picking out the most relevant synonyms and removing more unlikely ones. In addition to this, separate grammar files were made for each room in the game, making it possible to limit the amount of nouns recognizable in each room. For example the word "cat" is recognizable in the kitchen but not in the kitchen or bedroom.

3.3 Evaluation

3.3.1 User Testing

Users tested both versions of the game so that each of the score given could be compared. Each game starts with an introductory text, explaining how to play the game, the basic structure of commands and that the user should try using synonyms if stuck at any point. It also explains the goal and the name of all the rooms. This is the only thing the user is told before they start inputting commands. While they play they might only be given hints if the user is stuck at some point for quite a while. Examples of these hints are “speak clearer”, “try synonyms” or “input should be at least a verb and a noun”.

Before the user plays any game, they fill in a form asking for their personal data, how well they would rate their spoken and written english and if they have any speech impediment, see appendix A.1 figure A.2 and A.3. The user then plays one version of the game and after fills in the “System Usability Scale”-questionnaire detailed in 2.4. They then plays the other version and once again fills in the questionnaire. The game the user starts to play is varied between the users, so that almost half of the testers started with speech and half with text.

While the user plays the game records each command and how many commands is used. When the user is done this data gets saved to a file that is later done for analysis.

3.3.2 System Usability Scale

Using the System Usability Scale as described in 2.4, a score for each system is calculated. However, since all the odd questions are “positive statements” (for example “I think that I would like to use this system frequently”) while all the even questions are “negative statements” (such as “I found the system unnecessarily complex”) in nature, they have to be converted in order to make them work together. This is done as follows:

$$f_i = \begin{cases} 5 - Q_i, & \text{when i is even} \\ Q_i - 1, & \text{when i is odd} \end{cases} \tag{3.1}$$

Where Q_i is the answer to question numbered i. The total point is then:

$$2.5 * \sum_{i=1}^{10} f_i \tag{3.2}$$

The score lies in the range 0-100. A high score means that the system is easy to use and liked by the users, while a low score means that it should be improved before publishing. (Brooke, 1996)

Chapter 4

Results

In this section the results based on the performed user tests and user evaluation are presented. There were 9 test users, whereof 2 females and 7 males. All of them were 21-27 years old and were studying at KTH.

4.1 Effectiveness

4.1.1 Timescale

Table 4.1 declares the average time in seconds it took for the users to complete each version of the game. The data shows that it takes 38% more time to complete the speech version when compared to the text version.

Speech	Text
445.22	310.22

Table 4.1. Average time to complete the game

4.1.2 Commands

Table 4.2 declares the average amount of commands used before completing the game. The data shows that it takes almost double (99.67%) the amount of commands to complete the speech version when compared to the text version. This can also be seen in figure 4.1, where the average amount of commands used in each version of the game are compared to the ideal number of commands for each version. The ideal takes in consideration a realistic first playthrough of the game, where the player does not “magically know” what items are in each room. Therefore, when entering a room, the command "search room" was used to get the list of items located in the room. No items in the rooms were checked, just taken and used, unless the puzzle involved looking at an item (e.g. searching the desk to find the key). In figure 4.1 it is also shown that the ideal number of commands are pretty much the same for both versions of the game, making the difference in average number of commands even more significant.

Speech	Text
64.33	31.22

Table 4.2. Average number of commands used

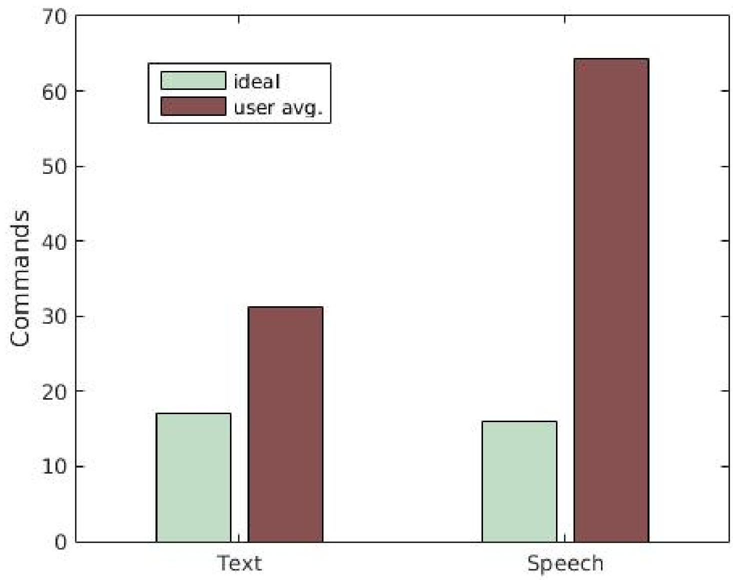


Figure 4.1. Average amount of commands used compared to the ideal

Figure 4.2 shows the number of commands used over time. As you would expect, the speech version follows the logic of "the longer time played the more commands used". However, this is not the case for the text version, where the time played seems irrelevant to the amount of commands used.

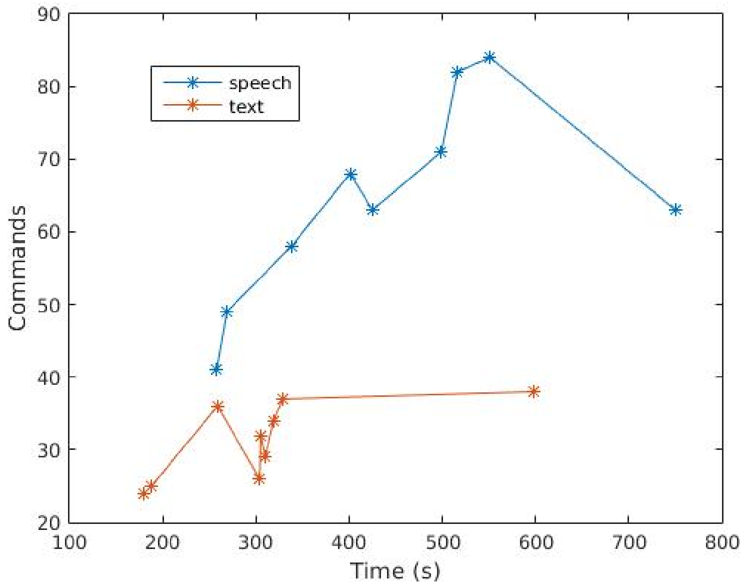


Figure 4.2. Number of commands used over time

4.2 Ease of Use

4.2.1 English Confidence

Each user rated their spoken and written English on a scale 1-5, where 1 equals “not good” and 5 equals “fluent”. The users were divided into groups based on their estimated English level and the average amount of commands, time played and usability score were calculated for each group. The results are presented in figure 4.3, figure 4.4 and figure 4.5. The average time for users of English level 4 did not differ much between versions, although for the users of English level 3 and 5 a more significant difference is shown.

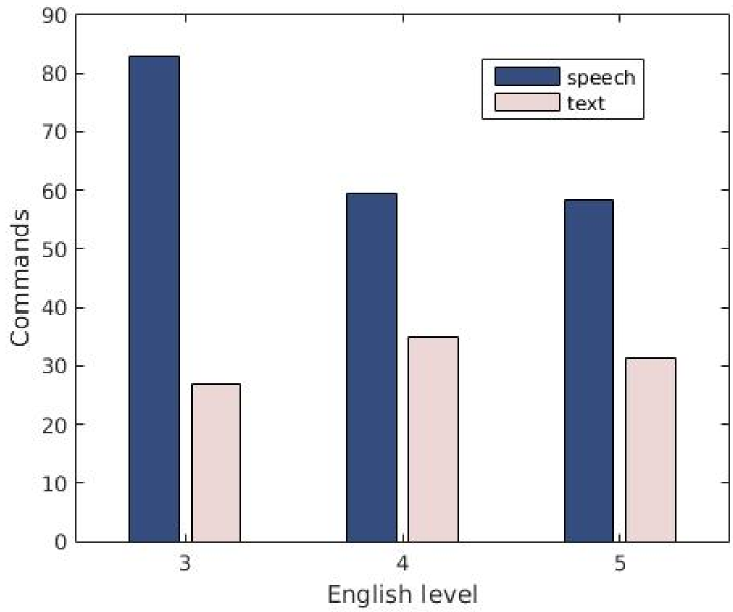


Figure 4.3. Average amount of commands used per English level-group

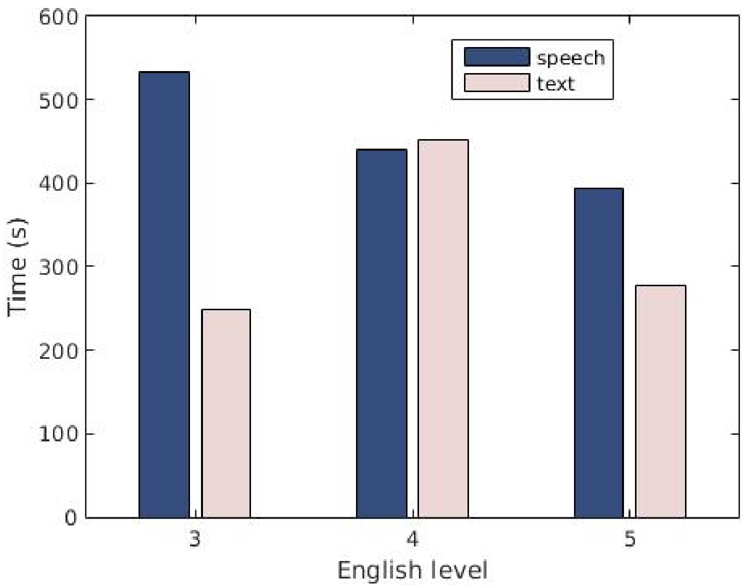


Figure 4.4. Average time to complete the game per English level-group

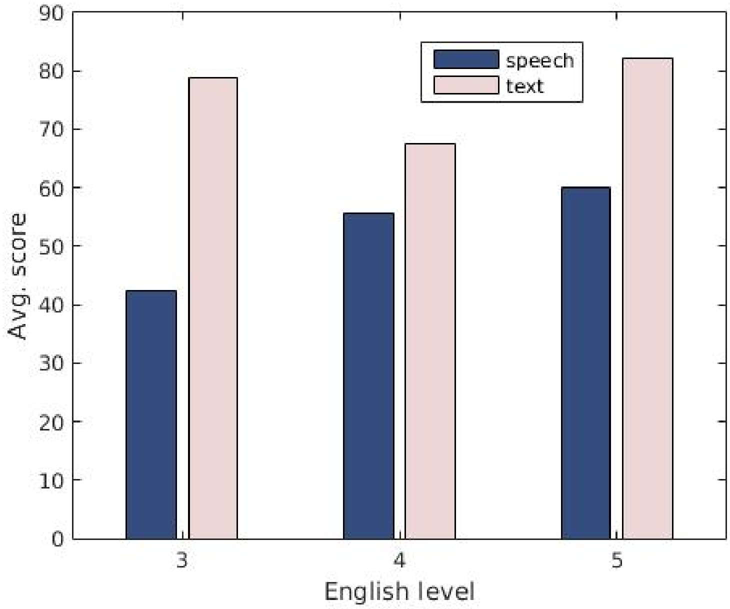


Figure 4.5. Average score based on the SUS per English level-group

4.3 Satisfaction (SUS)

Figure 4.6 shows the average score for each statement in the questionnaire described in section 2.4. The scores have been converted according to formula 3.1 in section 3.3.2.

The average total usability score based on the SUS as calculated in formula 3.2 for each version can be seen in table 4.3.

Speech	Text
54.17	78.06

Table 4.3. Average score based on the SUS

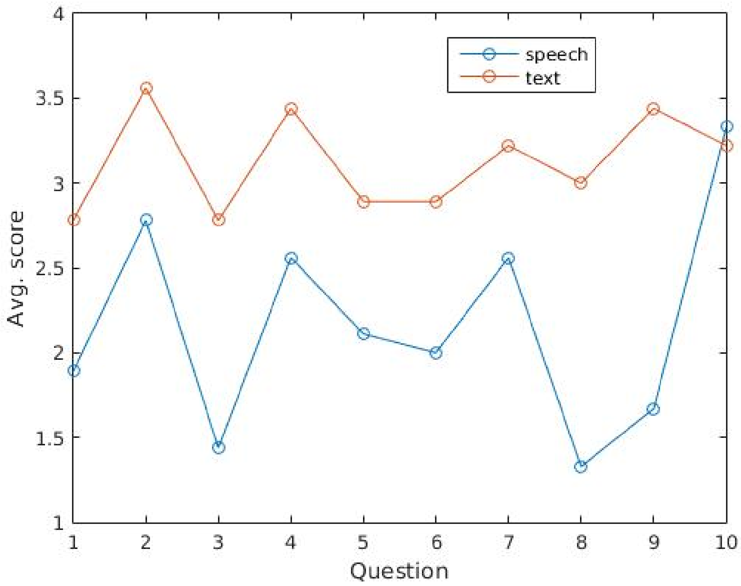


Figure 4.6. Average converted score for each statement

Chapter 5

Discussion

5.1 Comparison of Input Methods

After having tested both the text and speech version of the game and had the test users evaluate the different versions, the results show that text input is more effective than speech input. Even though it may consume more time typing a command than speaking it, the text version took less time overall and a fewer number of commands to complete. One reason behind this may be that the text version handles more synonyms for the relevant verbs than the speech version does, enabling a greater variation of the commands needed to progress through the game. Another reason may be the difference in how the user input can be registered. When speaking into a microphone there are many variations that can occur, such as different pronunciations, speaking volume, light or dark voice, etc., while typing on a keyboard have no such variations. Due to this the speech version have a higher risk of misunderstanding the user input and may force the user to repeat a command several times before the speech recognizer gets it right.

As seen in section 4.2.1, the level of confidence that the person has in their English may affect how they perform using the speech version. If a person think that they are fluent in English, they play for less time and use fewer commands. Figure 4.3 and 4.4 support this theory. This speaks for that speech as an input method could have uses, but before implementing it into a system an analysis of the English level of the user base should be done. If the system would be primarily used in native English speaking countries it might work equally to a system using text input.

When it comes to the System Usability Scale, the speech version rated on an average of 54.17 points and the text version on an average of 78.06 out of 100. Neither system is considered highly usable, but the text version has a significantly higher usability level than the speech version. Looking at figure 4.6 it can be seen that, with the exception of statement 10 where it is the same between the methods, the speech version is rated consistently worse than the text version, where speech is rated about 1-1.5 points less than text on statements 1-7 and statement 8 and 9 is rated about 2 points lower. This shows that in almost all aspects, speech needs improvement.

5.2 Sources of Error

The implementations of Sphinx4 and Stanford POSTagger might not live up their full capacity. There might be other or additional ways to implement them to make the game work even better. For example Sphinx4 does provide tools to train the recognizer, which might have made the speech recognition more accurate. Editing the implementation of Stanford POSTagger would presumably not affect the resulting difference between versions since it is implemented exactly the same for both versions of the game. Editing the implementation of Sphinx4, however, may cause changes in the results, since it is solely used in the speech version.

All test users had Swedish as their main spoken and written language, which might affect the accuracy of the speech version more than of the text version. When typing a command all that matters is grammar and spelling, but when speaking a command you also need the right pronunciation. It should also be noted that the users were asked to rate their own English level, so a personal bias could affect their rating. They might actually be better than they think or they might rate themselves higher than they should. If the users were to take some sort of test to get a more accurate ranking of their English level it would make the results more reliable.

The results are based on tests performed with 9 different users, which may not be enough to validate them. It would be appropriate to perform additional tests in order to strengthen the results. It might also be profitable to perform tests with users of different age groups and technical skills.

5.3 Future Research

If this project were to be enhanced in the future, it would be recommended to get a better way of ranking the user's english than having them do it themselves. A more varied user-group would also give a wider perspective: different ages, different technical knowledge and different pre-existing knowledge of similar applications. Most of the users in this experiment had played a text-based game before so they had some idea even before of what they were supposed to do.

Chapter 6

Conclusion

Text is better as input method for NLIs than speech in most cases. It takes less time to perform tasks needed, less tries to get the desired outcome and it has an overall higher usability level. However, if the user is confident in its English competence it will perform tasks with a higher correctness in the speech version. The user will also be more satisfied with it, at a level close to the text version. Given that the user base has English as it's native language, speech is a viable option as input method for NLIs and may be investigated further for that particular usage. For most situations and uses, however, text is much prefered.

Bibliography

Allen, J. F. (2003). *Natural language processing.* Published in Encyclopedia of Computer Science.

Brooke, J. (1996). *SUS: a 'Quick and Dirty' Usability Scale.* Redhatch Consulting Ltd.

Cambridge Dictionaries Online (2016). *Natural Language.* http://dictionary.cambridge.org/dictionary/english/natural-language.

Chowdhury, G. G. (2003). *Natural language processing.* American Society for Information Science and Technology.

CMU Sphinx (2015). *Sphinx 4.* http://cmusphinx.sourceforge.net/wiki/sphinx4:webhome.

Hendrix, G. G. (1982). *Natural-Language Interface.* SRI International, California.

ISO.org (1998). *Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts.* `https://www.iso.org/obp/ui/#iso:std:63500:en`.

Jones, K. S. (2001). *Natural Language Processing: A Historical Review.* Computer Laboratory, University of Cambridge.

Larsson, V. and Qvarfordt, J. (2015). *Taligenkänning som inmatningsmetod för naturligt språk.* Degree Project in Computer Science, KTH.

Lee, C.-H., Soong, F. K., and Paliwal, K. K. (1996). *Automatic Speech and Speaker Recognition, Advanced Topics.* Kluwer Academic Publishers.

Liddil, B. (1981). *Zork, The Great Undergroun Empire.* Published in Byte Magazine Volume 06 Number 02 - The Computer and Voice Synthesis.

Ribes, M. M. (2015). *Natural Language Interface Technology in Computer Games.* Degree-project for Computer Engineering.

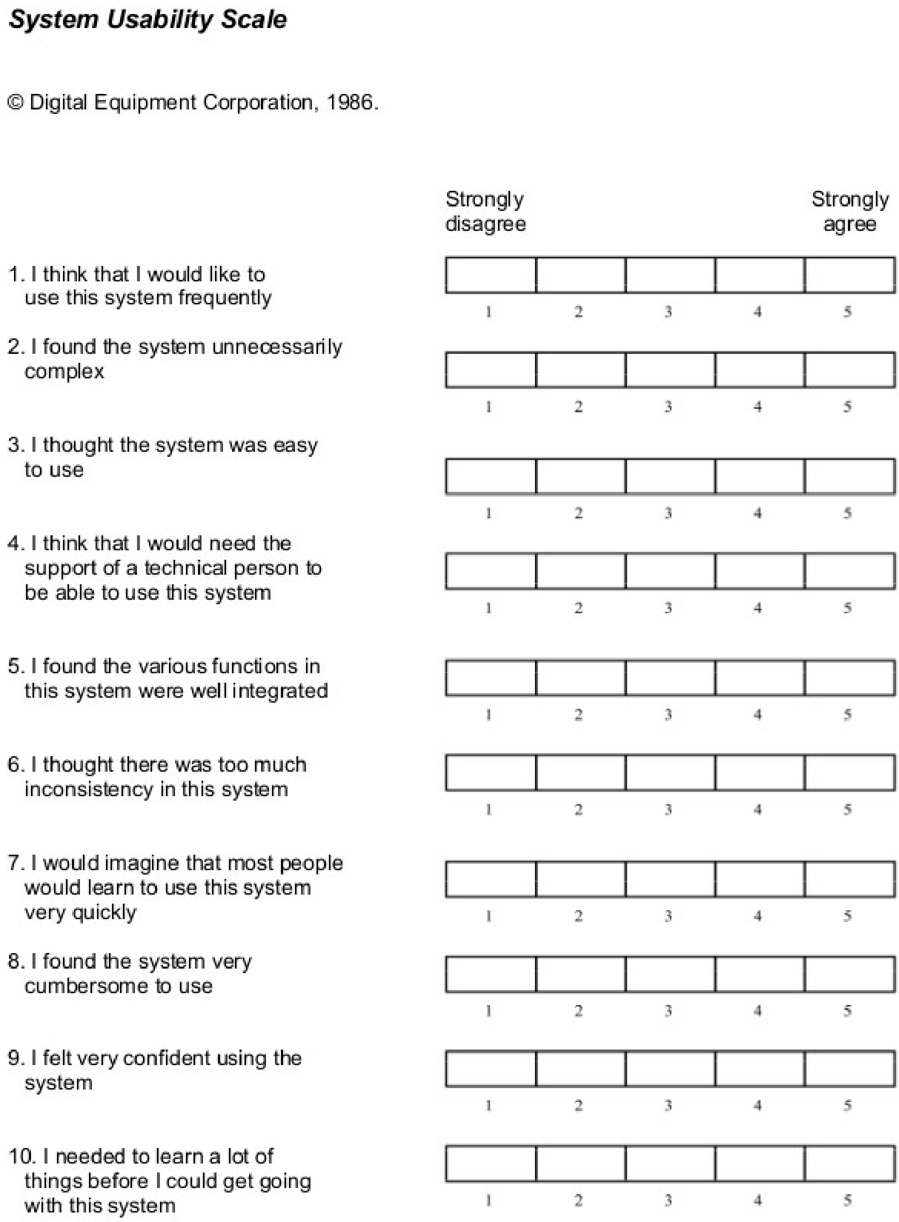
Sweetser, P. (2008). *Emergence in Games*. Published by Course Technology, a part of Cengage Learning.

The Stanford NLP Groups (2015). *Stanford Log-linear Part-Of-Speech Tagger*. http://nlp.stanford.edu/software/tagger.shtml.

Appendix A

Appendix

A.1 Questionnaires

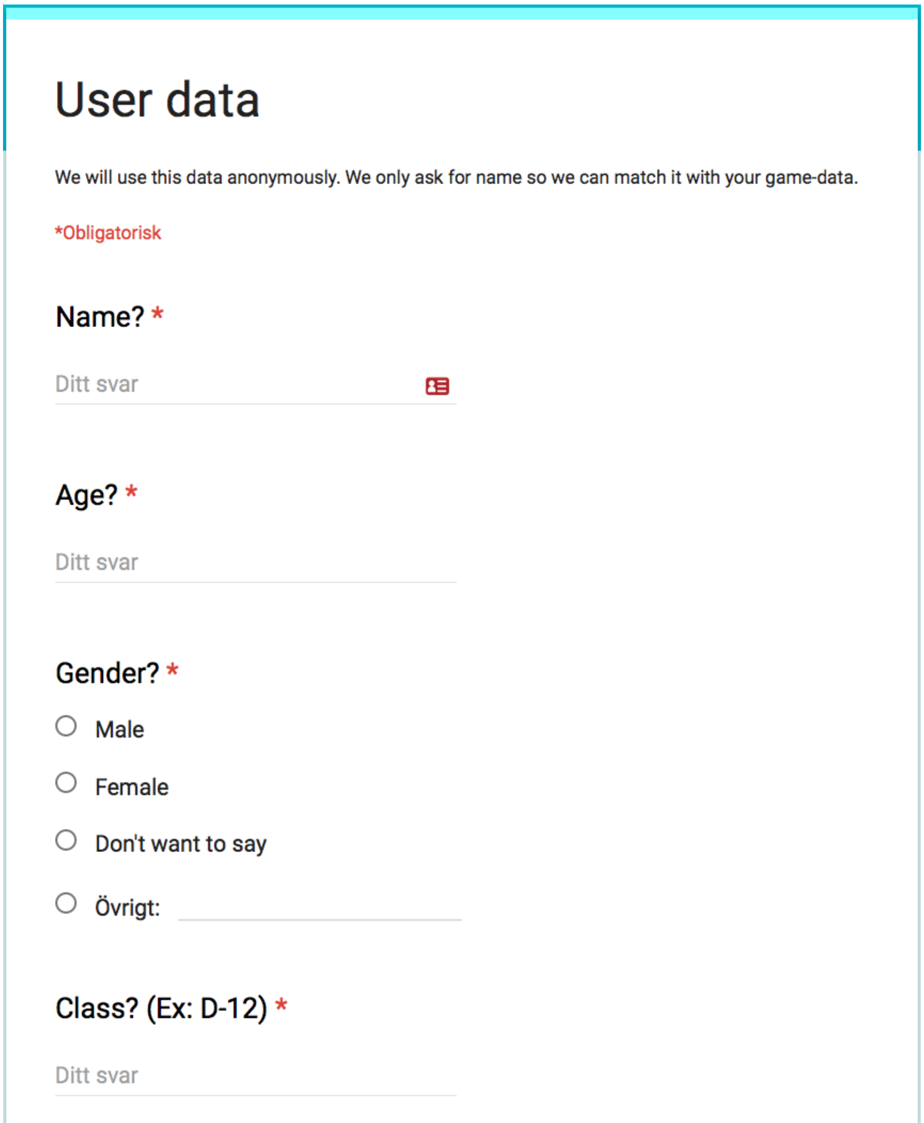


System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Figure A.1. Questions used in the System Usability Scale



User data

We will use this data anonymously. We only ask for name so we can match it with your game-data.

*Obligatorisk

Name? *

Ditt svar

Age? *

Ditt svar

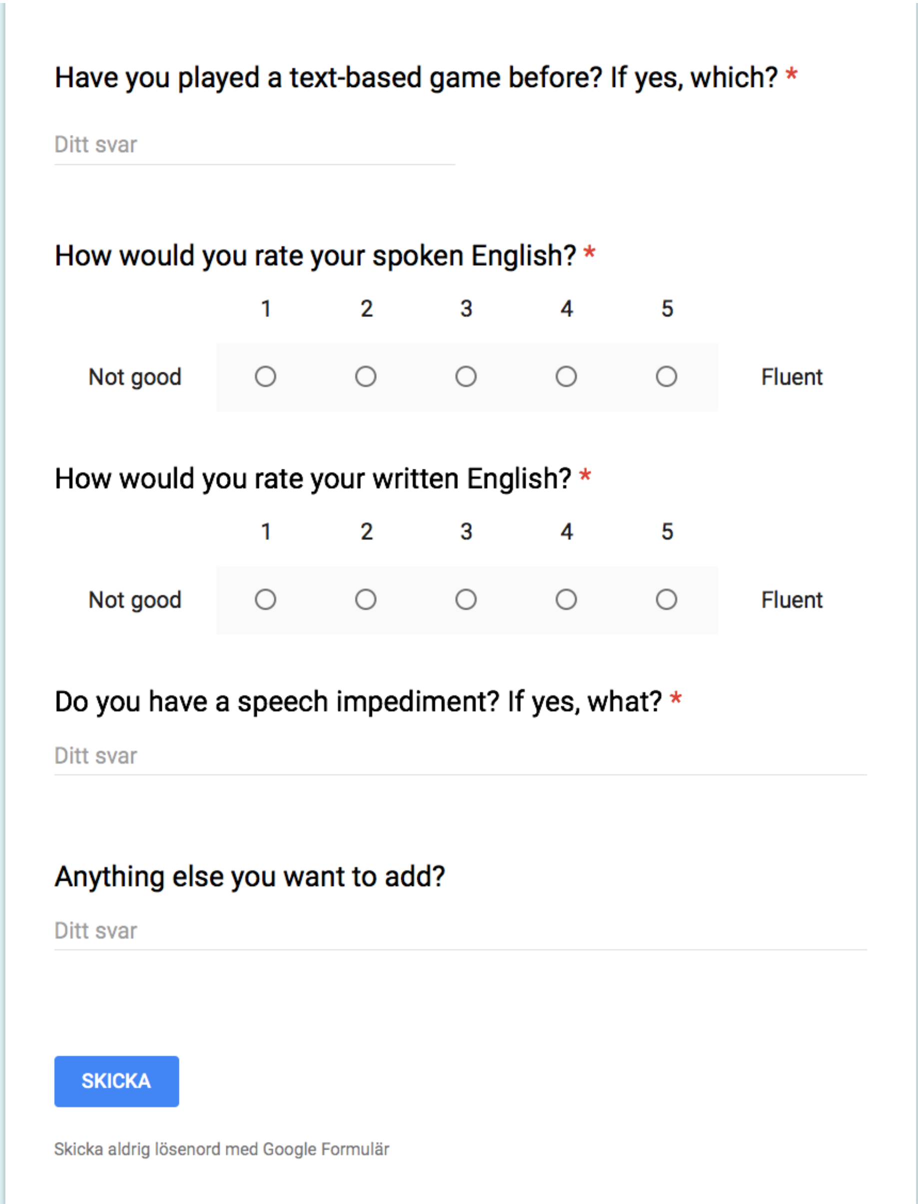
Gender? *

- Male
- Female
- Don't want to say
- Övrigt:

Class? (Ex: D-12) *

Ditt svar

Figure A.2. Part 1 of the questionnaire used to gather user data



Have you played a text-based game before? If yes, which? *

Ditt svar

How would you rate your spoken English? *

	1	2	3	4	5	
Not good	○	○	○	○	○	Fluent

How would you rate your written English? *

	1	2	3	4	5	
Not good	○	○	○	○	○	Fluent

Do you have a speech impediment? If yes, what? *

Ditt svar

Anything else you want to add?

Ditt svar

SKICKA

Skicka aldrig lösenord med Google Formulär

Figure A.3. Part 2 of the questionnaire used to gather user data