



DEGREE PROJECT, IN COMPUTER SCIENCE , FIRST LEVEL
STOCKHOLM, SWEDEN 2015

Taligenkänning som inmatningsmetod för naturligt språk

VICTOR LARSSON & JESPER QVARFORDT

KTH ROYAL INSTITUTE OF TECHNOLOGY

CSC SCHOOL



**KTH Computer Science
and Communication**

Taligenkänning som inmatningsmetod för naturligt språk

VICTOR LARSSON
JESPER QVARFORDT

Degree Project in Computer Science, DD143X
Handledare: Michael Minock
Examinator: Örjan Ekeberg

CSC, KTH 2015-05-08

Sammanfattning

Att interagera med ett system genom att använda naturligt språk görs genom att använda ett Natural Language Interface (NLI) som innebär svårigheter för både system och användare. Syftet med denna studie var att undersöka om taligenkänning är en användbar inmatningsmetod för naturligt språk baserat på ISO-definitionen av användbarhet. Detta gjordes genom att utveckla ett spel baserat på det klassiska textbaserade spelet Zork där användaren ger kommandon i naturligt språk som sedan användes för en användbarhetsstudie. Studien utfördes med mellangrupsdesign med två grupper av användare där den ena gruppen använde tangentbordsinmatning och den andra taligenkänning med Web Speech API. Både kvalitativ och kvantitativ data insamlades under testet om användarnas effektivitet, beteende och uppfattning av systemet. Resultaten visade att taligenkänning hade lägre effektivitet med avseende på tidsåtgång och antal fel men upplevdes mer positivt än tangentbordsinmatning av deltagarna. Därför kan taligenkänning anses vara en rimlig inmatningsmetod då effektivitet inte är en avgörande faktor.

Abstract

Speech recognition as input method for natural language

To interact with a system using natural language is done by using a Natural Language Interface (NLI) and means difficulties for both users and the system. The purpose of this study was to examine whether speech recognition is a useful input method for natural language based on the ISO definition of usability. This was done by developing a game based on the classic text-based game Zork, where the user enters commands in natural language, that was then used to conduct a usability test. The test used between-group design on two groups of users where the first group used keyboard input and the second group used speech recognition with Web Speech API. Both qualitative and quantitative data was gathered during the test regarding the users efficiency, behaviour and perception of the system. 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. Speech recognition can thereby be considered a viable input method in cases where efficiency is not of crucial significance.

Innehåll

1	Introduktion	1
1.1	Inledning	1
1.2	Syfte	2
1.3	Problemformulering	2
2	Bakgrund	3
2.1	Natural Language Interface (NLI)	3
2.1.1	Natural Language Processing (NLP)	4
2.1.2	Zork - The Great Underground Empire	5
2.2	Taligenkänning	6
2.2.1	Tidigare forskning	6
2.2.2	Taligenkänningsystem	8
2.3	Systemutvärdering ur användbarhetsperspektiv	8
3	Metod	11
3.1	Verktyg och programspråk	11
3.2	Spelet	12
3.3	Pilottest	13
3.4	Deltagare	13
3.5	Procedur	13
3.5.1	Introduktion	13
3.5.2	Användning	14
3.5.3	Avslutning	14
3.6	Mätningar	14
3.6.1	Kvantitativ data	14
3.6.2	Kvalitativ data	15
3.7	Plats och material	15
4	Resultat	17
4.1	Effektivitet	17
4.1.1	Tidsåtgång	17
4.1.2	Antalet effektiva och totala antalet kommandon	19
4.2	Fel vid inmatning	21

4.3	System Usability Scale (SUS)	21
4.4	Observationer och intervju	21
5	Diskussion	23
5.1	Jämförelse av inmatningsmetoderna	23
5.1.1	Effektivitet	23
5.1.2	Användarnas uppfattning	24
5.1.3	Användarnas beteende	25
5.2	Metodanalys	25
5.2.1	Felkällor	25
5.3	Framtida forskning	26
5.4	Slutsats	26
6	Referenser	27

Kapitel 1

Introduktion

1.1 Inledning

Ett system som har ett Natural Language Interface (NLI) innebär att en användare interagerar med systemet med hjälp av naturligt språk. Detta kan anses vara det mest flexibla gränssnittet för kommunikation mellan människa och dator (Barabás & Kovács, 2015). Användaren matar in naturligt språk till datorn där inmatningen ofta sker med tangentbord som tolkas som kommandon, vilka utförs av datorn. Detta skiljer sig ifrån andra användargränssnitt som till exempel använder knappar som användaren klickar på för att få datorn att utföra kommandon.

Tangentbordsinmatning har med de begränsningar som funnits i teknologin varit den enklaste och vanligaste inmatningsmetoden. Med dagens teknologi har många interaktionsätt utvecklats som utöver att optimera inmatningsmetod beroende på situation även tillåter personer med funktionsnedsättning att använda datorer. I många av dagens system är tangentbord inte längre det självklara valet av inmatningsmetod. Detta kan ses på mobiltelefoner där det är vanligt att se inmatningsmetoder som till exempel fingergester och taligenkänning.

Ett klassiskt exempel på ett system som använder NLI som har funnits länge är det textbaserade spelet Zork. I Zork använder spelaren naturligt språk för att ge kommandon till datorn om vad karaktären i spelet ska göra. Zork utvecklades redan 1977 och introducerade ett nytt sätt att interagera med ett spel.

Tidig forskning inom ämnet behandlar olika sätt att mata in kommandon till ett system. Forskningen fokuserades kring ett specifikt problem så som att mata in en sträng siffror i ett system med taligenkänning för att effektivisera en viss arbetsuppgift.

Denna uppsats ämnar att undersöka användbarheten för tangentbords- och taligenkänningsinmatning för ett NLI-baserat system med fokus på den definition av användbarhet angiven i ISO 9241-11 (Brooke, 1996). Där definieras användbarhet som användarens:

- förmåga att slutföra uppgifter och kvaliteten av tillvägagångssättet
- konsumerade tillgångar för att utföra uppgifter
- subjektiva reaktioner till användningen av systemet.

1.2 Syfte

Syftet med uppsatsen är att undersöka hur en användares sätt att interagera med ett NLI-baserat system påverkas av inmatningsmetoden med fokus på användarens upplevelse och effektivitet. De jämförda inmatningsmetoderna är tangentbordsinmatning och taligenkänning i ett enkelt utforskande spel implementerat med ett NLI. Under utvecklingen av detta spel kommer Zork användas som inspiration för struktur och det naturliga språk användaren kommer att använda vid interaktion med systemet.

Till grund för denna undersökning ligger den bristande forskningen kring hur olika inmatningsmetoder med NLI-baserade system påverkar användarna. Eftersom ett NLI-baserat system använder naturligt språk för att beskriva vilka instruktioner som ska utföras är det värdefullt att undersöka om tal kan användas effektivt.

Av den förberedande förstudien framkom inga tidigare arbeten som behandlar skillnaderna mellan de två inmatningsmetoderna i den ovan beskrivna kontexten (NLI). Målet med uppsatsen är att presentera argument om taligenkänning är ett rimligt substitut till tangentbordsinmatning för NLI-baserade system.

1.3 Problemformulering

- Är taligenkänning ett rimligt substitut för tangentbordsinmatning som inmatningsmetod för NLI-baserade system?
- Påverkas användarnas beteende gentemot systemet beroende på inmatningsmetoden?
- Påverkas användarnas uppfattning av systemet beroende på inmatningsmetoden?

Kapitel 2

Bakgrund

Denna del av rapporten utforskar tidigare arbeten samt de nödvändiga verktygen för att utveckla det spel som ska användas under studien. Undersökningsområdena är NLI-baserade system, hur de utformats genom historien, hur de fungerar och hur de använts samt taligenkänning och hur det kan påverka användares prestation och upplevelse. Under avsnittet för Natural Language Processing analyseras spelet Zorks sätt att använda naturligt språk som inmatning för att utveckla det egna systemet.

2.1 Natural Language Interface (NLI)

Ett NLI är ett gränssnitt vars uppgift är att acceptera användares kommandon formulerade i naturligt språk och översätta dessa till programanrop som en dator kan förstå och utföra (Barabás & Kovács 2015). Med naturligt språk menas att användaren formulerar en mening uppbyggd av till exempel substantiv och verb för att kontrollera datorn. Nyckelordsbaserad NLI kan beskrivas som den enklaste metoden som enbart kräver reguljär grammatik för att förstå språket (Barabás & Kovács, 2015). Redan 1950 nämnde Alan Turing konceptet att en människa ska kunna kommunicera med naturligt språk med en maskin. Han föreslår ett test i sin rapport "Computing Machinery and Intelligence" (Turing, 1950). Detta test är idag känt som Turingtestet och används för att avgöra om en maskin kan imitera mänsklig intelligens. Turing menar att om en människa efter en konversation med en maskin inte kan avgöra om det är en maskin eller människa, så har maskinen uppfyllt kriteriet för mänsklig intelligens (Barabás, 2013).

Tidiga prototyper av NLI-baserade system började synas redan under slutet av 60-talet och början av 70-talet, varav en av de mest kända är SHRDLU, ett NLI utvecklat av Terry Winograd under 1968-1970 som lät användaren interagera med datorn via naturligt språk för att flytta och namnge virtuella objekt (Winograd, 1971). SHRDLU gav goda resultat, vilket resulterade i ökad optimism bland forskare (Barabás, 2013). Till en början var antalet NLI-baserade system litet, men redan under slutet av 70-talet kunde man se en tydlig ökning. Bland dessa fanns det

tidigare nämnda textbaserade spelet Zork.

Dagens NLI-baserade system har blivit betydligt mer avancerade än de som utvecklades under 70-talet (Barabás & Kovács, 2015). Framför allt har de till en större grad integrerats i människors vardagliga liv. Ett exempel är Apples virtuella personliga assistent Siri, en applikation för operativsystemet iOS som är utvecklat för mobila enheter. Siri använder sig dessutom utav taligenkänning istället för tangentbordsinmatning som många tidigare kända NLI-baserade system.

Fördelar med ett NLI är att användaren inte behöver lära sig ett artificiellt språk för att utföra kommandon vilket är enklare för vanliga användare (Androustopoulos, Ritchie & Thanisch, 1995). Det är dock viktigt att poängtera att de system som används i praktiken använder ett begränsat språk, vilket resulterar i att en inlärningsperiod är nödvändig för användarna. På grund av det naturliga språkets komplexitet krävs en förenklad modell. Det betyder att problem kan uppstå då en användare formulerar en mening med exempelvis en negation som kanske inte stöds. Användare kan få föreställningen att systemet är intelligent, det vill säga kan härleda fakta. Författarna menar att mycket arbete krävs för att konfigurera ett NLI för att vara av tillräckligt god kvalitet.

Winograd (1972) beskriver problemet med syftning då naturligt språk används. När människor kommunicerar används kunskap om omgivningen för att bland annat bestämma vad en mening syftar på. Därför menar Winograd att en mycket viktig del av ett NLI är att datorn kan avgöra vilket subjekt som användaren syftar på. Ett annat problem är att sammanhanget är mycket viktigt för en korrekt tolkning då ett ord i naturligt språk kan ha olika betydelser i olika sammanhang. I engelskan är till exempel ordet *break* både ett verb för “att bryta, slå sönder eller bromsa”, men också substantivet för “rast och broms”. Detta gör att datorn måste kunna känna igen och kunna hantera prepositioner.

2.1.1 Natural Language Processing (NLP)

Användarens inmatning i naturligt språk behandlas i ett Natural Language Processing (NLP) system. Ett NLP-system processar inmatning på olika nivåer beroende på tillämpning. Dessa nivåer utgör en modell till hur ett språk kan beskrivas (Barabás, 2013). Nivåerna består av en fonologisk, morfologisk, lexikalisk, syntaktisk, semantisk, pragmatisk samt en samtalsnivå. De vanligaste NLP-systemen implementerar enbart de nivåer som behandlar meningar och ord utan att analysera betydelse mer ingående. Dessa nivåer innefattas av de morfologiska, lexikaliska, syntaktiska och semantiska nivåerna eftersom dessa behandlar naturligt språk genom att analysera den textsträng som inmatas.

Barabás (2013) beskriver den fonetiska nivån som nivån för tal då det används som inmatning och behandlar ords uttal. Den morfologiska nivån behandlar ord och deras olika former som till exempel ett ords form i plural. För att kunna avgöra hur ett ord ska tolkas utför den lexikaliska nivån en analys som bestämmer vilken ordklass ordet tillhör. Till exempel avgör den lexikaliska analysen om ett ord är ett verb eller ett substantiv. Semantiken i meningen som matats in bestäms av den

2.1. NATURAL LANGUAGE INTERFACE (NLI)

semantiska nivån. Då ord kan ha olika semantiska tolkningar är det viktigt att denna nivå beslutar vilken som ska användas. Den pragmatiska nivån har i syfte att ge en underliggande mening av texten. Detta förklarar författaren är en komplex analys eftersom en stor mängd kunskap behövs om den värld som systemet behandlar. Samtalsnivån behandlar meningar i ett större perspektiv för att till exempel avgöra vad ett pronomen som till exempel "han" syftar på. Denna nivå behandlar också hur ord kan tillhöra ett objekt som till exempel en sida kan tillhöra en viss bok.

I ett annat arbete beskriver Barabás och Kovács (2015) de tidiga NLP-systemen som en förenklad modell av den Barabás (2013) presenterat som användes för att kontrollera en robot med hjälp av naturligt tal. De moduler de använde för att styra roboten var en texttolk, morfologi- och semantikanalys.

2.1.2 Zork - The Great Underground Empire

Detta avsnitt behandlar spelet Zork som NLP-system för att belysa de aspekter som är viktigast vid utvecklingen av det spel som används i studien av inmatningsmetoder.

Zork spelas vanligtvis med att genom tangentbordsinmatning beskriva kommandon i naturligt språk för att sedan få ett svar i form av text från systemet (Anderson, Blank & Lebling, 1979). Grundtanken var att simulera den riktiga världen tillräckligt bra för att få spelaren att spenderade sin tid på att lösa problem i spelet, snarare än att förstå spelet. Det är viktigt att spelet har ett brett ordförråd så att spelaren inte behöver undra huruvida denne använder rätt ord eller inte.

Författarna förklarar att den underliggande strukturen till Zork består av en databas med fyra större datatyper: rum, objekt, handlingar och aktörer. Instanser av dessa typer kan innehålla olika egenskaper och funktioner som gör instansen unik. Ett träd kan till exempel ha egenskapen *old* för att särskilja det från andra träd i skogen.

Zork körs i en slinga som utför ett antal steg. Dessa steg är i den absolut enklaste formen som följer:

- användaren matar in ett kommando som programmets texttolk behandlar
- kommandot utförs
- omgivningen (rummet, objekt i närheten) behandlas.

Syftet med Zorks texttolk är att tolka och reducera de kommandon som användaren anger till lågnivåkommandon som programmet kan förstå. Denna process börjar genom att verifiera att de ord som inmatats även finns i programmets ordförråd. Tolken försöker sedan bland dessa hitta en handling och ett eller två olika objekt. Därefter fastställs vilken handling och vilka objekt användaren avser referera. Det kontrolleras att alla objekt som hänvisats till är tillgängliga genom att objektet måste vara i användarens egendom eller finnas i rummet. Om flera objekt har samma namn används angivna adjektiv för att avgöra vilket av objekten som hänvisas

till, till exempel *rusty knife*. Om inget adjektiv angivits så frågar tolken vilket av objekten användaren menade. Därefter kontrolleras att alla objekt som krävs för att utföra handlingen har blivit hängivna. Om objekt saknas så försöker tolken att hitta ett indirekt objekt - till exempel ett objekt som nyligen använts - för att slutföra handlingen. Om tolken inte kan hitta ett indirekt objekt så ber den spelaren att ange rätt objekt, exempelvis "Attack troll with what?" om inget vapen har angetts.

När tolkningen av kommandot är klart så kan programmet försöka att utföra handlingen. När handlingen utförs så anropas först funktioner på objekt som kan ha inverkan på handlingens utfall. Objekten som dessa funktioner utförs på är följande:

- Aktören som utför handlingen. Alla aktörer kan inte utföra alla handlingar.
- Fordonet som aktören befinner sig i. Fordon (eller avsaknad av fordon) kan begränsa möjlighet till en viss rörelse.
- Handlingen. Alla handlingar kan inte utföras i kombination med alla objekt.
- Fordonet. Fordonet anropas återigen för att kunna agera efter att möjliga förändringar har utförts då handlingen anropades.
- Rummet spelaren befinner sig i. Alla handlingar kan inte utföras i alla rum.

Var och en av dessa funktioner anropas i ordning och vid varje funktion finns en möjlighet att handlingen hanteras av funktionen. Om en funktion väljer att hantera handlingen så anropas inte resterande funktioner. Funktioner kan även skriva ut meddelanden utan att påverka handlingen i sig. Om ingen av funktionerna väljer att hantera handlingen så skrivs "nothing happened" ut. Detta bör emellertid inte hända för de flesta rimliga kommandon då stor vikt har lagts vid att skapa rimlig respons för många kommandon (Anderson, Blank & Lebling, 1979).

2.2 Taligenkänning

Taligenkänning (eng. speech recognition) innebär att en användare interagerar med ett system med hjälp av att tala. Många tillämpningar av taligenkänning använder sig dock inte utav naturligt språk, utan korta kommandon eller enkel vokabulär som lättare känns igen av ett taligenkänningssystem. Detta har använts av både Hauptmann och Rudnicky (1990) samt Martin (1989) i deras undersökningar. I ett arbete av Chapanis (1975) drogs slutsatsen att kommunikation i form av tal kan vara det mest värdefulla sättet att kommunicera vid problemlösande situationer som också stärks i arbetet av Martin (1989).

2.2.1 Tidigare forskning

Det finns flera dokumenterade fördelar med att använda taligenkänning. Martin (1989) beskriver hur personer blev effektivare i situationer där mer än en uppgift

2.2. TALIGENKÄNNING

utförs samtidigt. Vid taligenkänningsinmatning är fler responskanaler tillgängliga och uppgifter fördelas på de olika sinnen. Martin undersökte också två vanligt förekommande påståenden om att använda taligenkänning som interaktion med en dator: det är effektivare än att använda tangentbord och ger en ytterligare kanal för respons. Experimentet utfördes genom att låta sju personer använda mus, kommandon inmatade med tangentbord eller genvägskommandon med tangentbord för att utföra ett antal uppgifter genom att rita fyrkanter på skärmen som de sedan kunde ta bort eller färga. Antalet utförda uppgifter mättes tillsammans med antalet klick eller tangenttryck. Experimentet delades upp i två faser, där testpersonerna i första fasen fick använda taligenkänning som komplement till de givna interaktionsmetoderna. Resultatet av undersökningen bekräftar påståendena då 63% av uppgifterna gjordes då taligenkänning användes till skillnad mot 32% av uppgifterna då taligenkänning inte användes.

Annan tidigare forskning som beskriver hur taligenkänning presterar bättre än tangentbordsinmatning har utförts av Hauptmann och Rudnicky (1990). Experimentet gick ut på att 18 testpersoner med tre olika inmatningsmetoder skulle mata in siffror. Inmatningsmetoderna som användes taligenkänning, taligenkänning med tangentbordskorrektur och enbart tangentbordsinmatning. Siffrorna som skulle matas in visades på skärmen under första delen av experimentet och gavs sedan på papper för den andra delen av experimentet. Resultatet visade att taligenkänning är ett snabbt sätt att mata in siffror på och författarna beskriver hur längre och mer komplex inmatning kan bevisa ytterligare hur taligenkänning kan vara ett bättre inmatningsalternativ. Dessutom drogs slutsatsen att taligenkänning för sifferinmatning har stor fördel för vanliga användare då taligenkänning eliminerar problemet att hitta tangenterna på tangentbordet.

Hauptmann och Rudnicky (1990) förklarar att tidigare forskning har delade meningar angående tangentbordinmatning och taligenkänning och hävdar att problem med kvaliteten på taligenkänningssystemen kan vara en faktor. Ytterligare forskning stärker dessutom denna argumentation (Martin, 1989). Båda rapporterna tar upp hur taligenkänningssystemens felfaktor försvårar slutsatsen. Martin tar upp hur taligenkänning i optimala förhållanden ofta ger bra resultat och motsatsen i en mer naturlig miljö.

Kaljurand och Alumäe (2012) ger exempel på hur tal skiljer sig från skrift och de problem detta medför. De engelska orden *cite*, *sight* och *site* tas upp som exempel på ord som inte kan skiljas då de talas. Förkortningar som *three gee* bör behandlas som 3G och *nineteen ninty* som 1990 och avsaknad av denna funktionalitet kan utgöra problematik. Författarna nämner också hur ord som uttalas liknande utan definierbar paus emellan är problematiskt som *I scream* och *ice cream*. Den sista komplikationen författarna nämner är att skiljetecken så som punkter och komma-tecken inte kan verbaliseras. De sammanfattar dessa problem som fyra konventioner där svårigheter uppstår i tal men inte i skrift:

- Då ord uttalas likadant men har olika betydelse
- Då förkortningar uttalas
- Då olika ordföljder uttalas likadant
- Då skiljetecken ska markeras med tal.

2.2.2 Taligenkänningsystem

För att kunna utföra en grundlig undersökning av inmatningsmetoder krävs kunskap om system för taligenkänning som finns på marknaden så att en slutsats kan dras om vilket system som passar detta arbete. Nedan följer de tillgängliga och mest relevanta taligenkänningsystemen.

CMU Sphinx är samlingsnamnet för en grupp olika taligenkänningsystem som utvecklats för Java på Carnegie Mellon University. De system som ingår i gruppen kallast Sphinx 1-4 samt PocketSphinx som är ett system för mobila enheter. Systemen samlar tillsammans 20 år av universitetsforskning men körs dock enbart på Linux (CMU Sphinx, 2015).

Web Speech API är ett API (Application Programming Interface) för javascript som släpptes av Google med Chrome version 25. Web Speech API har fördelen att det fungerar utan komplicerad konfiguration och tillhandahåller både taligenkänning som inmatningsmetod för olika dialekter. Detta API är förhållandevis nytt då det släpptes 2012 (Shires & Wennborg, 2012).

2.3 Systemutvärdering ur användbarhetsperspektiv

System Usability Scale (SUS) är en standardiserad metod för att kunna få jämförbar data om ett systems användbarhet (Brooke, 1996). SUS baseras på ett frågeformulär där 10 frågor är ställda i en specifik ordning där varannan fråga hänvisar till en positiv respektive en negativ aspekt. Användare svarar på frågorna enligt en skala från 1-5 (*strongly disagree* till *strongly agree*) vanligen direkt efter användaren testat det system som utvärderas. Ett system med ett SUS-resultat över 68 anses som godkänt.

2.3. SYSTEMUTVÄRDERING UR ANVÄNDBARHETSPERSPEKTIV

SUS-resultatet beräknas genom att tolka svaren från 0-4, vilket ger:

$$V_i = \begin{cases} 5 - S_i & \text{då } i \text{ är jämn} \\ S_i - 1 & \text{då } i \text{ är udda} \end{cases} \quad (2.1)$$

där S_i betecknar svaret på fråga i . Deltagarens SUS-resultatet blir då

$$2.5 \cdot \sum_{i=1}^{10} V_i \quad (2.2)$$

Då flera användare testat ett system används medelvärdet för SUS-resultaten och ger bra resultat även för lågt deltagande. Grunden till SUS ligger i definitionen av användbarhet angiven i ISO 9241-11 (Brooke, 1996). De frågor som SUS innehåller tar upp dessa aspekter och kan användas för de flesta system (Brooke, 2006). Problem som dokumenterats med metoden är att användare som inte har engelska som modersmål har svårt med ord som *cumbersome* och hur de tolkar *very* eller *strongly*. Därför bör formuläret fyllas ut under tillsyn av en ansvarig person som kan assistera (Finstad, 2006). Enligt Nielsen (1993) är endast 15 deltagare nödvändigt för att hitta alla användbarhetsproblem och endast 5 deltagare för att finna ungefär 80 % av alla användbarhetsproblem.

Kapitel 3

Metod

För att besvara problemformuleringen används användartester för att insamla både kvalitativ och kvantitativ data om användarnas effektivitet och uppfattning. Mellangruppsdesign har använts som experimentdesign för att kunna jämföra och analysera resultat mellan två grupper. Ena gruppen använde tangentbord som inmatningsmetod medan den andra gruppen använde taligenkänning som inmatningsmetod. Detta utfördes eftersom den enda förändrade variabeln i undersökningen då är inmatningsmetoden. Dessutom minimeras inlärningseffekter (Sallnäs Pysander, 2015).

Gruppen med tangentbordsinmatning behandlas som kontrollgrupp. Båda grupperna innehöll lika många deltagare. Spelet som deltagarna interagerar med och de uppgifter deltagarna utför är identiska mellan grupperna för att kunna göra jämförelser av de två inmatningsmetoderna med avseende på:

- deltagarnas effektivitet
- deltagarnas beteende gentemot systemet
- deltagarnas uppfattning och upplevelse av systemet
- svårigheter för användaren
- svårigheter för systemet.

Systemet utvärderades för båda inmatningsmetoderna med fokus på skillnader i användarnas effektivitet och upplevelse. Ingen kompensation gavs ut till deltagarna.

3.1 Verktyg och programspråk

För att undersöka inmatningsmetoderna utvecklades ett textbaserat spel inspirerat av det NLI-baserade spelet Zork. Detta motiveras i att Zork är ett spel som använder ett enkelt men mycket kraftfullt NLI. Spelets utforskande karaktär tillåter att

undersöka användares beteende gentemot systemet för tangentbordsinmatning och taligenkänning.

Web Speech API används som taligenkänningsystem. Testet kan då utföras som en webbaserad applikation, utvecklat med Javascript, vilket gör både utveckling och tester kan göras lättare på grund av tillgänglighet och enkel konfiguration.

Enligt egna tester av Web Speech API klarar tjänsten av att urskilja de tre nödvändigaste av de fyra skriftliga konventioner som inte återspeglas i tal som nämns av Kaljurand och Alumäe (2012). Web Speech API kan skilja på ord med nära identiskt uttal beroende på det sammanhang ordet används i. De meningar som testades var: “The report cite that the findings were correct”, “This is the site for the construction of the house” och “When I close my eyes I lose my sight”. I dessa exempel är det *cite*, *site* och *sight* som uttalas likadant. Web Speech API kan också skilja på ordföljder som låter likadant. För att testa detta användes meningarna “The ice cream was tasty” och “I scream from the top of my lungs”. I detta exempel är det *ice cream* och *I scream* som har likande uttal. Dessutom kan Web Speech API omvandla uttalade siffror och sifferkombinationer till skrivna siffror. Till exempel blev både “nineteen hundred eighty four”, “one thousand nine hundred and eighty four” samt “nineteen eighty four” tolkat som 1984 av systemet.

Meningarna ändrades i realtid då ordets kontextuella betydelse blev tydligare. Till exempel kunde ett ord börja att skrivas ut som *site* men ändras till *cite* på grund av att sammanhanget blev tydligare desto längre meningen var. Den sista konventionen som nämnts av Kaljurand och Alumäe (2012) som Web Speech API inte klarade av var att placera ut skiljetecken. Detta medför emellertid inga komplikationer då spelets inmatning inte är i behov av denna språkfunktionalitet.

Web Speech API uppvisar sofistikerad funktionalitet och användes därför som taligenkänningsystem.

3.2 Spelet

Spelet som utvecklas har samma utseende som en typisk kommandotolk och använder samma engelska språkstruktur som Zork. För inmatning av text skriver användaren in ett kommando i naturligt språk och trycker på returtangenten. Ett kommando betyder härmed en inmatning i naturligt språk. Vid inmatning genom taligenkänning håller användaren ned mellanslag under tal. Kommandot matas in då tangenten släpps. Inga möjligheter för redigering av kommandot kan göras vid inmatning med taligenkänning på grund av komplexiteten detta skulle innebära för användaren.

Spelet utspelar sig i ett universitet och är utformat att vara enkelt att slutföra genom att ha ledande instruktioner som svar vid ett kommando om vad användaren förväntas göra härnäst. Hela användarens väg genom spelet kan alltid ses på skärmen och sparas ned efter spelet slutförts.

Spelet innehåller tre underliggande uppgifter. Dessa uppgifter krävs för att slutföra spelet men är inte avslöjade för deltagaren utan används som delmoment i ana-

3.3. PILOTTEST

lysen av deltagarens användning. Deluppgifterna är att hämta en tom flaska, ge en fylld flaska till den törstiga karaktären och slutligen att slutföra spelet. Uppgifterna kräver olika sorters kommandon där den andra uppgiften kräver flest kommandon för att slutföra.

3.3 Pilottest

Innan metoden fastställdes utfördes två pilottester för att utvärdera både testet och spelet som deltagarna använde. Ett pilottest för varje inmatningsmetod gjordes. Pilottesten resulterade i att spelet modifierades så att det blev enklare för användarna att förstå de steg som krävdes för att slutföra spelet genom att göra beskrivningar mer ledande. Tidtagning lades till i systemets bakgrund för att deltagarna inte ska bli påverkade och stressade av att testet innefattar tidtagning. För att undvika oavsiktlig inmatning vid taligenkänning ändrades taligenkänningen från kontinuerlig inmatning till att deltagaren måste hålla in mellanslag för att tala in kommandon. Utöver detta förblev proceduren oförändrad.

3.4 Deltagare

16 deltagare användes i undersökningen varav två personer användes för pilottest. Eftersom spelet kräver att användaren läser text på engelska användes inte deltagare med språk- eller talsvårigheter eller låga kunskaper i engelska. Deltagarna rekryterades till största del från Kungliga Tekniska Högskolan. Förutom deltagare från Kungliga Tekniska Högskolan rekryterades andra deltagare med lägre datorvana. Deltagarna valdes ut efter ett bakgrundsformulär där deltagaren får fylla i uppgifter om datorvana och tidigare erfarenheter med NLI-baserade system och taligenkänning. Detta gjordes för att strategiskt placera deltagaren i grupperna så att deltagarnas bakgrunder för inmatningsmetoderna ska vara likgiltiga och därmed jämförbara. För varje inmatningsmetod användes 2 personer med lägre datorvana och 5 personer med hög datorvana. Alla deltagare hade svenska som modersmål med goda engelskkunskaper. Varje grupp innehöll jämt fördelade deltagare som använt taligenkänning tidigare och som spelat ett textbaserat spel tidigare.

3.5 Procedur

Proceduren för inmatningsmetoderna är identiska. Testsessionen består av en inledning, användning och avslutning.

3.5.1 Introduktion

Testet inleds med att deltagaren får en introduktion om vad testet handlar om. Alla instruktioner läses från ett manus för att säkerställa att samma instruktioner ges till varje testdeltagare. Användaren ska redan ha fyllt ett bakgrundsformulär

som getts ut vid ett tidigare rekryteringstillfälle och en inmatningsmetod för denne deltagare har bestämts. Inmatningsmetoden bestäms utifrån de tidigare deltagarna och hur de placerats i grupperna för att göra grupperna jämna. Deltagaren fyller i ett medgivande att ljud och bild spelas in under testet för att fortsätta.

Deltagaren får ett instruktionsblad innehållande grundläggande kommandon för spelet. Deltagaren får sedan testa kommandon givna av testmoderatoren för att förstå systemet och inmatningsmetoden. De kommandon deltagaren får testa är ett enkelt kommando, ett synonymt kommando och två kommando innehållande en additiv konjunktion (*and* och *then*) och ett långt kommando innehållande två fullständiga meningar sammansatta av en konjunktion. Deltagaren får ställa eventuella frågor om systemet innan sessionen går vidare.

3.5.2 Användning

Testmoderatoren ber deltagaren att starta spelet och uppgiften att slutföra spelet visas på skärmen. Observationer och antalet korrekt ändrade kommandon antecknas i en anteckningsmall av testmoderatoren.

3.5.3 Avslutning

Vid slutfört spel får deltagaren fylla i ett SUS-formulär om hur de upplevde spelet som system. Detta utförs för att undersöka om deltagare upplever systemet annorlunda beroende på inmatningsmetod.

Efter att deltagaren fyllt i formuläret intervjuas denne för att få subjektiva åsikter om användningen utifrån ett antal ledande frågor. Testmoderators observationer av deltagarens användning av systemet kompletteras också med eventuella frågor om hur deltagaren tänkte eller resonerade. Frågorna ställdes på ett sätt så att de uppmuntrade deltagaren till att högt reflektera kring dennes uppfattning och tankar kring systemet.

Vid slutförd intervju avslutas testsessionen och spelets logg sparas ned för senare analys och de tillgängliga resultaten förs in i ett protokoll.

3.6 Mätningar

För att besvara problemformuleringen och tillhandahålla data angående systemets effektivitet och upplevelse görs utvalda mätningar.

3.6.1 Kvantitativ data

- Mätning av totala antalet fel gjorda av användare vid tangentbordsinmatning räknas av spelet då deltagaren redigerar ord. Flera redigeringar av samma ord och kommando räknas. Till detta adderas antalet felaktigt inmatade ord som räknas i spelets logg. Då användaren ändrar ett korrekt kommando till ett annat räknas dessa bort och antecknas av testmoderatoren under testets gång.

3.7. PLATS OCH MATERIAL

- Mätning av totala antalet fel gjorda av taligenkänningsystemet vid taligenkänningsinmatning mäts genom att räkna antalet felaktigt inmatade ord i spelets logg.
- Mätning av antalet effektivt gjorda kommandon av användare räknas i spelets logg och definieras som antalet kommandon som spelet uppfattat korrekt och inte innehåller fel. Upprepningar av kommandon på grund av felaktig tolkning av taligenkänningsystemet eller felstavningar vid tangentbordsinmatning räknas inte med som effektiva kommandon. Om deltagaren under testet använder ett kommando som inte stöds av spelets språk men uppfattas som ett rimligt kommando anses kommandot korrekt och testmoderatoren ger deltagaren det korrekta kommandot från spelets språk för att fortsätta. Detta görs också för att undvika att deltagaren fastnar på grund av begränsningar i spelet. Dessa två kommandon räknas sedan som ett. Detta mäts för att få en uppfattning om antalet kommandon användarna försökt att utföra.
- Mätning av antalet kommandon för att slutföra varje deluppgift räknas i spelets logg.
- Mätning av tidsåtgången för att lyckas med deluppgifter skrivs ut av spelet i webbläsarens logg och hämtas efter slutfört test.
- Mätning av uppfattning av användbarhet görs genom att låta deltagaren fylla i ett formulär enligt SUS.

För att undersöka statistisk signifikans användes Mann-Whitney U test. Detta test används eftersom normalfördelning inte kan säkerställas och en icke-parametrisk metod därför krävs.

3.6.2 Kvalitativ data

- Observation av beteende och uttryck från användaren under testsession antecknas under testets gång
- Intervju med användare efter testet ger subjektiva åsikter om upplevelsen.

Formulär behandlades och hanterades i Google Forms.

3.7 Plats och material

Testen utförades i olika lokaler under kontrollerade former. De lokaler som användes hade liknande akustik och endast testdeltagaren och testmoderatorer var närvarande i rummet. Deltagaren interagerade med systemet via en bärbar dator körandes Windows 8.1 i ett inkognitofönster i webbläsaren Google Chrome. Testtillfället filmades för att tillåta analyser i efterhand. Vid taligenkänningsinmatning användes en huvudmonterad mikrofon riktad mot munnen (Steelseries Siberia V2).

Kapitel 4

Resultat

I detta avsnitt visas resultat som jämförelser mellan de två inmatningsmetoderna. Mätningar av användarnas förmåga att slutföra uppgifter, benämnt effektivitet, visas genom mätningar av tidsåtgång, hur deltagare använt kommandon och antal fel vid inmatning. Slutligen presenteras data om användarnas upplevelse och uppfattningar av systemet.

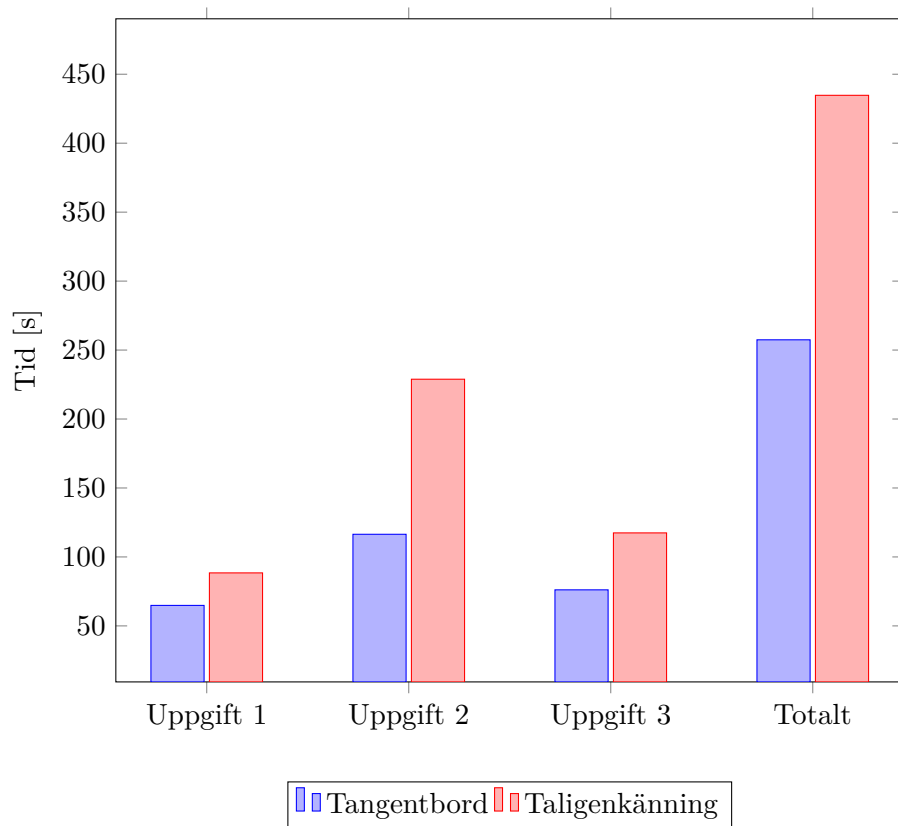
4.1 Effektivitet

4.1.1 Tidsåtgång

Tabellen nedan redovisar den genomsnittliga tiden det tog för testdeltagare att slutföra respektive uppgifter för de båda inmatningsmetoderna. Tiden anger antalet sekunder för varje deluppgift och den totala tiden för att slutföra alla uppgifter presenteras. Tidsåtgången för totaltiden skiljer signifikant mellan inmatningsmetoderna (Mann–Whitney $U=2$, $n_1 = n_2 = 7$, $p < 0.05$ två sidigt).

Tabell 4.1. Jämförelse mellan medelvärde \bar{t} för tider uppmätta för användare att utföra uppgifter där σ betecknar standardavvikelsen.

Tangentbord			Taligenkänning		
Uppgift	\bar{t}	σ	Uppgift	\bar{t}	σ
1	64.86 s	32.73 s	1	88.43 s	55.18 s
2	116.43 s	26.02 s	2	228.86 s	105.87 s
3	76.14 s	28.56 s	3	117.43 s	27.31 s
Totalt	257.43 s	66.42 s	Totalt	434.71 s	152.95 s



Figur 4.1. Diagram över medelvärdet för tidsåtgången för deluppgifter och totaltid.

4.1. EFFEKTIVITET

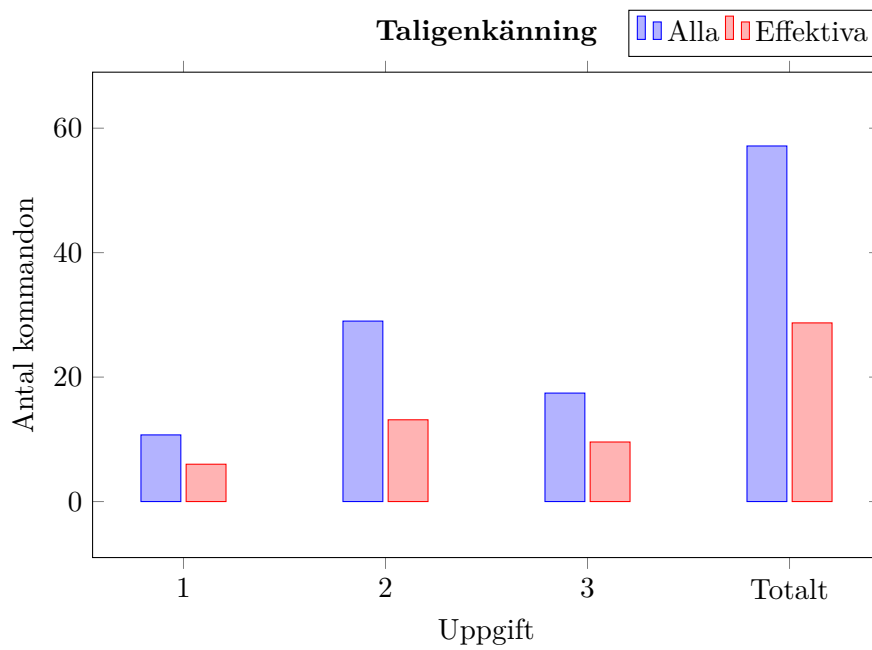
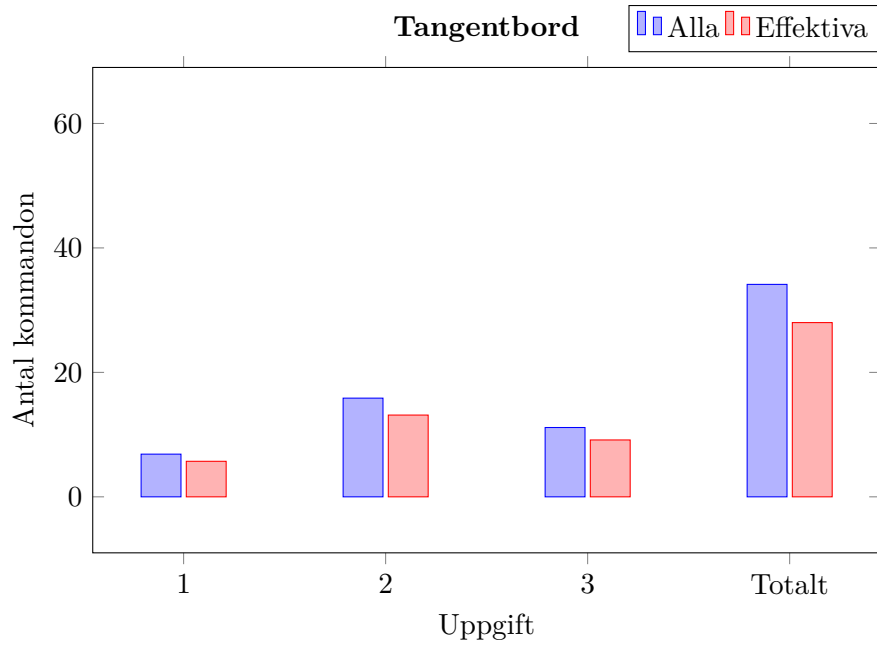
4.1.2 Antalet effektiva och totala antalet kommandon

Deltagarnas användning av kommandon presenteras här. Medelvärden för totala antalet kommandon (betecknat alla), effektiva kommandon och skillnaden mellan dessa presenteras. Differansen mellan det totala antalet och effektiva kommandon skiljer sig signifikant mellan inmatningsmetoderna (Mann-Whitney $U=1$, $n_1 = n_2=7$, $p<0.05$ två sidigt).

Tabell 4.2. Jämförelse mellan antal kommandon för att utföra uppgifter där Δ betecknar differensen.

Tangentbord			
Uppgift	Alla	Effektiva	Δ
1	6.86	5.71	1.14
2	15.86	13.14	2.71
3	11.14	9.14	2
Totalt	34.14	28	6.14

Taligenkänning			
Uppgift	Alla	Effektiva	Δ
1	10.71	6	4.71
2	29	13.14	15.86
3	17.43	9.57	7.86
Totalt	57.14	28.71	28.43



Figur 4.2. Diagram över antalet kommandon för tangentbordsinmatning och taligenkänning för jämförelse.

4.2 Fel vid inmatning

Nedan presenteras de fel som uppmätts. Medelvärde för det totala antalet fel visas tillsammans med standardavvikelsen. Skillnaden mellan inmatningsmetodernas fel är inte statistiskt signifikanta (Mann–Whitney $U=10$, $n_1 = n_2 = 7$, $p < 0.05$ två sidigt).

Tabell 4.3. Jämförelse av fel vid inmatning där \bar{e} betecknar medelvärdet för fel och σ standardavvikelsen.

Inmatningsmetod	\bar{e}	σ
Tangentbord	11.71	6.47
Taligenkänning	27.57	18.94

4.3 System Usability Scale (SUS)

Nedan presenteras hur deltagarna uppfattade spelets användbarhet på en skala från 0-100.

Tabell 4.4. Jämförelse av SUS-värde för inmatningsmetoder.

Tangentbord	Taligenkänning
78.33	79.69

4.4 Observationer och intervju

I detta avsnitt presenteras resultat av de observationer och den intervju som har gjorts av testmoderatorerna under användartesterna. Utmärkande drag i användares sätt att interagera med systemet har observerats och antecknats samt användares generella beteende under testet. Nedan presenteras en sammanställning av de återkommande och utmärkande observationer som gjorts tillsammans med de sammanställda svaren av intervjufrågorna. Genom att filma testsessionerna har material analyserats även i efterhand.

- De deltagarna som använde tangentbordsinmatning kände sig självsäkra och majoriteten observerades inte ha några problem med inmatningen. De upplevdes mer målinriktade än deltagarna som använde taligenkänningsinmatning som upplevdes lugnare.
- Samtliga deltagare kände sig bekväma med systemet, oberoende av inmatningsmetod.
- Ingen skillnad mellan datorvana och mindre datorvana deltagare iaktogs för båda inmatningsmetoderna.

- 28,57% deltagare som använde tangentbordsinmatning upplevdes ha problem med stavning av vissa engelska ord.
- Ingen av deltagarna som använde tangentbordsinmatning nämnde eller observerades påverkas av felaktig inmatning som redigerades.
- Bland de som använde taligenkänning var det däremot 57,14% som upplevdes ha problem med att taligenkänningssystemet ofta tolkade ord fel. Vid de flesta tillfällena var det samma kommando som gav upphov till fel och andra kommandon kunde inmatas felfritt.
- 42,9% av deltagarna som använde taligenkänning observerades tycka att tolkningsfel av systemet var roliga.
- Bland de deltagare som använde taligenkänning så nämnde samtliga att spelet var roligt och observerades visa mer entusiasm än vid tangentbordsinmatning. Av de som använde tangentbordsinmatning nämnde 42,86% detta.
- Inga av de deltagare som använde sig av tangentbordsinmatning kände sig frustrerade över systemet. Detta går att jämföras med taligenkänningsgruppen där 28,57% av deltagarna kände frustration över att systemet inte tolkade deras kommandon rätt. Dessa deltagare hade aldrig använt ett NLI-baserat system tidigare.
- Av de deltagare som använde taligenkänning upplevde 85,71% att tiden gick fort då spelet testades. För deltagarna som använde tangentbordsinmatning ansåg 71,43% detta.
- Samtliga deltagare som använde taligenkänning kände att de skulle kunna tänka sig att använda ett NLI-baserat system i sin vardag efter testet. 57% av deltagarna som använde tangentbordsinmatning skulle kunna tänka sig detta.
- Huvudproblemet med taligenkänning observerades vara felaktig tolkning av systemet som ledde till upprepade inmatningar.

Kapitel 5

Diskussion

5.1 Jämförelse av inmatningsmetoderna

5.1.1 Effektivitet

Tangentbordsinmatning är effektivare för NLI-baserade system enligt resultaten. Då båda inmatningsmetoderna har liknande fördelning för uppgifterna så är taligenkänning mer tidskrävande och fler fel gjordes (tabell 4.1, 4.3 och figur 4.1). Taligenkänningen hade högre standardavvikelse i alla mätningar för totala värden, vilket indikerar att deltagarnas prestation skiljer sig mer vid taligenkänning. Enligt Martin (1989) är taligenkänning ett tidsmässigt effektivare sätt att interagera, vilket motsäges. Skillnaden i tidsåtgång kan bero på det höga antalet fel som taligenkänningen uppvisade som beror av att taligenkänningsystemet tolkar användarens kommandon felaktigt. Antalet effektiva kommandon visar ingen större skillnad mellan inmatningsmetoderna. Däremot är antalet totala kommandon mycket högre för taligenkänning. Detta indikerar att deltagare använder fler resurser per effektivt kommando vid taligenkänning.

Detta stärker de slutsatser som Hauptmann och Rudnicky (1990) och Martin (1989) presenterat; taligenkänningens prestation är till stor del beroende av taligenkänningsystemets kvalitet. Trots att denna undersökning utfördes under kontrollerade förhållanden visade sig taligenkänning ge markant fler fel än tangentbordsinmatning vilket indikerar att naturligt språk kräver taligenkänningsystem av hög kvalitet. Det kan antas att felantalet hade varit ännu högre i en vardaglig miljö med mer brus. Dock är skillnaden i antalet fel inte statistiskt signifikant, vilket gör att slutsatsen om detta inte går att säkerställa.

Bristen på redigeringsmöjligheter vid taligenkänning är också en möjlig faktor till den höga tidsåtgången. Vid tangentbordsinmatning kan ett fel redigeras så fort deltagaren upptäckt det. För en van datoranvändare är tidsåtgången för att redigera ett tangentbordsinmatat kommando liten. Detta skiljer sig från taligenkänning där hela kommandon måste utföras på nytt vid ett felaktigt kommando, oberoende datorvana eller erfarenhet. Att låta användare redigera ett intalat kommando med tangentbordet före inmatning gör att taligenkänningens syfte försvinner i ett NLI-

baserat system, men kan vara en möjligt förbättring. Varje enskilt fels stora inverkan på effektiviteten vid taligenkänning och det högre antalet fel ger emellertid ett möjligt argument för att taligenkänning i dagens läge inte är praktiskt användbart med avseende på effektivitet då Web Speech API används.

Ett problem för användaren som uppkommer vid lågpresterande taligenkänningsystem är att uttalet blir viktigt. Ofta gjordes ett antal fel i följd för samma kommando då taligenkänningsystemet tolkade deltagarens kommando fel. Deltagaren kunde därför fastna på enbart ett kommando men ha lågt felantal på andra kommandon. För att taligenkänningsystemet ska vara användbart i praktiken krävs att det kan klara av felaktigt uttal. Detta skulle kunna göras med en definierad grammatik som taligenkänningsystemet kan jämföra tvetydiga kommandon mot. Detta gör att andra taligenkänningsystem kan ge bättre effektivitet.

En fördel med taligenkänning är att då taligenkänningsystemet tolkar deltagarens kommando korrekt är stavningen alltid rätt. Användare behöver heller inte leta efter rätt tangenter vilket kan anses som mer naturligt som också nämns av Hauptmann och Rudnicky (1990). Deltagare som har problem med stavning kan därför ha fördel av taligenkänning, givet att felantalet är lågt.

5.1.2 Användarnas uppfattning

Effektivitetsanalysen indikerar att taligenkänning är mindre effektivt både tidsmässigt och för antalet fel. Detta återspeglades dock inte i användarnas uppfattning av systemet.

Enligt resultaten från SUS-formuläret (tabell 4.4) anser deltagarna att taligenkänningsinmatning är minst lika användbart som tangentbordsinmatning, vilket inte borde vara fallet med anseende på de uppmätta resultaten i effektivitet. Detta kan bero på att alla deltagare som använde taligenkänningsinmatning nämnde att spelet var roligt och majoriteten ansåg även att spelet tog kort tid att slutföra. Det kan visa att deltagare är mer förlåtande mot fel som uppstår vid taligenkänning. Majoriteten av de deltagare som använt tangentbordsinmatning ansåg också att spelet tog kort tid att slutföra, men endast 42,86 % uttryckte att spelet var roligt.

Deltagarna som använde taligenkänningsinmatning kunde alla tänka sig att använda ett NLI-baserat system i sin vardag till skillnad från deltagarna som använde tangentbordsinmatning. Att NLI-baserade system uppskattas av deltagarna med taligenkänning är tydligt. Detta skulle kunna förklaras med att ett NLI-baserat system är mer naturligt att interagera med genom tal än att använda tangentbordsinmatning. Det kan antas vara relaterat till att naturligt språk upplevs mer naturligt att tala än att skriva vilket är oberoende datorvana. Detta beskrivs även av Androutopoulos, Ritchie och Thanisch (1995) där naturligt språk inte kräver ett artificiellt språk, vilket kan vara lättare för vanliga användare.

5.2. METODANALYS

5.1.3 Användarnas beteende

Genom att studera antalet effektiva kommandon visades att deltagarna använde likvärdigt antal kommandon för båda inmatningsmetoderna eftersom antalet effektiva kommandon var mycket lika (tabell 4.2 och figur 4.2). Däremot observerades att deltagare som använde tangentbordsinmatning verkade mer inriktade på att slutföra spelet till skillnad mot taligenkänning där flera deltagare uppvisade ett lugnare beteende. Detta kan bero på att deltagarna som använde taligenkänning upplevdes förbereda sig mer för varje inmatning.

Som effektivitetsanalysen visade var taligenkänning mindre effektivt både tidsmässigt och för antalet fel. Detta observerades både som positivt och negativt då nära hälften av deltagarna med taligenkänningsinmatning ansåg att felet kunde anses roliga. Däremot då felet upprepade sig flertal gånger visade deltagare frustration som också togs upp i intervjun. Dessa deltagare hade dock aldrig spelat ett NLI-baserat spel vilket kan förklara deras beteende. Tangentbordsinmatningens redigeringsmöjligheter hjälpte deltagare vid inmatning och samma frustration som uppkom vid taligenkänning är således inte möjlig till samma utsträckning.

Datorvana hade enligt observationer ingen påverkan på användningen och alla deltagare ansågs sig vara bekväma med systemet. Detta kan stärka argumentet att naturligt språk är enkelt att använda för vanliga användare (Androutsopoulos, Ritchie & Thanisch, 1995). Dock krävs en mer utförlig undersökning med fler mindre datorvana deltagare för att vidare stärka detta.

5.2 Metodanalys

Vid testsessionerna visades att samma användbarhetsproblem uppkom redan då 3 deltagare utfört testerna för båda inmatningsmetoderna. Detta beskrivs av Nielsen (2000) och stärker argumentet att relativt få deltagare uppvisar de vanligaste användbarhetsproblemen. Fler deltagare ger bättre statistisk säkerställning men antalet deltagare som deltog anses tillräckligt. Ett problem med mellangrupsdesign är att fler deltagare krävs än till exempel inomgruppsdesign och den utförliga rekryteringsprocessen gjorde att fler deltagare inte användes. En alternativ experimentdesign som kunnat användas är inomgruppsdesign för att låta alla deltagare prova på båda systemen. Dock har denna design en stor nackdel då inläringseffekter uppkommer för att ge tillräckligt tillförlitliga resultat vilket var grunden till att mellangrupsdesign användes.

5.2.1 Felkällor

Det hade varit önskvärt att rekrytera deltagare med engelska som modersmål men visade sig vara svårt. Det kan antas att resultaten för taligenkänning kan ha blivit annorlunda om detta hade lyckats. Dock anses alla deltagare ha talat tillräckligt bra engelska för att detta inte ska ha påverkat resultaten i någon betydande utsträckning.

Eftersom observationer och subjektiva åsikter behandlas, finns risk för att testmoderatorer kan ha förbisett viktiga observationer. Denna felkälla har reducerats genom att videoinspela testsessionerna för att i efterhand kunna återanalysera deltagaren.

Deltagare kan vid intervjuer ha utelämnat för undersökningen relevant information. Dessutom kan testsessionen ha upplevts som en utsatt position vilket kan ha medfört att deltagare inte har känt sig bekväma.

5.3 Framtida forskning

På grund av taligenkänningsystemets betydande faktor kan andra taligenkänningsystem ge andra resultat. Denna rapport drar därför endast slutsatser kring Web Speech API. Framtida forskning om andra taligenkänningsystem utförda enligt samma metod som i denna undersökning kan ge ytterligare insikter kring taligenkänningsens användbarhet som inmatningsmetod. Undersökningar med fler deltagare, framför allt med mindre datorvana, kan ge bredare förståelse till hur datorvana kan påverka användningen.

För att förbättra effektiviteten vid taligenkänning bör möjligheten att kunna redigera inmatning undersökas. Om ett effektivt redigeringsätt kan utvecklas kommer detta rimligtvis att förbättra effektiviteten hos taligenkänning som inmatningsmetod.

5.4 Slutsats

Taligenkänning är inte lika effektivt som tangentbordsinmatning för naturligt språk. Användarnas uppfattning var dock mer positiv gentemot systemet då taligenkänning användes som inmatningsmetod. Användarna upplevdes mer förlåtande gentemot systemet samt ansåg att systemet i sin helhet var roligare. Detta kan anses visa att användarnas uppfattning och beteende gentemot systemet påverkas av inmatningsmetoden som används.

Slutsatsen kan därför dras att taligenkänning med Web Speech API är ett rimligt substitut för tangentbordsinmatning för naturligt språk i de fall där effektivitet inte är en avgörande faktor.

Kapitel 6

Referenser

Anderson, T. A., Blank, M. S., Lebling, D. P., 1979. Zork: A computerized Fantasy simulation game. [pdf] Tillgänglig på: <<http://www.computer.org/csdl/mags/co/1979/04/01658697.pdf>> [Hämtad 16 februari 2015]

Androutsopoulos I, Ritchie G. D., Thanisch, P., 1995. Natural Language Interfaces to Databases – An Introduction. [online] Tillgänglig genom: Cornell University Library <<http://arxiv.org/abs/cmp-lg/9503016>> [Hämtad 15 februari 2015]

Barabás, P., László K., 2015. Grammar Representation Forms in Natural Language Interface for Robot Controlling. Emergent Trends in Robotics and Intelligent, [online] Volume 316, s. 65-72. Tillgänglig genom: Springer Link <<http://link.springer.com>> [Hämtad 15 februari 2015]

Barabás, P., 2013. Domain- and language-adaptive natural language controlling framework [pdf] Tillgänglig på: <http://193.6.1.94:9080/JaDoX_Portlets/documents/document_15100_section_11933.pdf> [Hämtad 17 februari 2015]

Brooke, J. 1996. SUS - A quick and dirty usability scale. Usability evaluation in industry, 198, s.4-7. Tillgänglig genom: Google Books <<http://books.google.com>> [Hämtad 26 mars 2015]

Chapanis, A. 1975. Interactive Human Communication. Scientific American, 232, s. 36-42. Tillgänglig genom: MIT Media Lab <<http://web.media.mit.edu/>> [Hämtad 17 februari 2015]

CMUSphinx, 2015. About CMUSphinx. Pittsburgh: CMUSphinx. Tillgänglig på: <<http://cmusphinx.sourceforge.net/wiki/about>> [Hämtad 19 februari 2015]

Finstad, K. 2006. The System Usability Scale and Non-Native English Speakers. Journal of usability studies, 1(4), s. 185-188. Tillgänglig på: <<https://uxpa.org/>>

[sites/default/files/finstad_sus_non_native_speakers.pdf](#) > [Hämtad 26 mars 2015]

Hauptmann, A. G., Rudnicky, A. I., 1990. A comparison of speech and typed input [pdf] Tillgänglig på: <<http://acl-arc.comp.nus.edu.sg/archives/acl-arc-090501d3/data/pdf/anthology-PDF/H/H90/H90-1045.pdf>> [Hämtad 16 februari 2015]

Kaljurand, K., Alumäe, T., 2012. Controlled Natural Language in Speech Recognition Based User Interfaces. Lecture Notes in Computer Science, 7427, s.79-94. Tillgänglig genom: Springer Link <<http://link.springer.com/>> [Hämtad 19 februari 2015]

Martin, G. L. 1989. The utility of speech input in user-computer interfaces. International Journal of Man-Machine Studies [online] Volym 30 (4). s. 355-375. Tillgänglig genom: Science Direct <<http://www.sciencedirect.com/>> [Hämtad 16 februari 2015]

Nielsen, J., Landauer, T., K., 1993. A Mathematical Model of the Finding of Usability Problems. In Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems, s. 206-213. Tillgänglig genom: AMC <<http://dl.acm.org/>> [Hämtad 31 mars 2015]

Sallnäs Pysander, E., 2015. Diskussion om metoder och statistiska modeller. [email] (Personlig kommunikation, 11 mars 2015)

Shires, G., Wennborg, H., 2012. Web Speech API Specification. W3C. Tillgänglig på: <<https://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html>> [Hämtad 19 februari 2015]

Turing, A., 1950. Computing machinery and intelligence. Mind [online] s. 433-460. Tillgänglig genom: Jstor <<http://www.jstor.org/stable/pdf/2251299.pdf?acceptTC=true>> [Hämtad 19 februari 2015]

Winograd, T., 1971. Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. MIT AI Technical Report 235, Tillgänglig genom: MIT Libraries <<http://dspace.mit.edu/>> [Hämtad 16 februari 2015]

Winograd, T., 1972. Understanding Natural Language. Cognitive Physiology [online] Volym 3, s. 1-191. Tillgänglig genom: Science Direct <<http://www.sciencedirect.com/>> [Hämtad 19 februari 2015]

