An Approach for Anonymous Spelling for Voter Write-Ins Using Speech Interaction

by

Shaneé Terese Dawkins

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 18, 2009

Keywords: Universal Accessibility, E-Voting, User Interfaces, Word Prediction

Approved by

Juan E. Gilbert, Chair, Professor of Computer Science and Software Engineering
Cheryl Seals, Associate Professor of Computer Science and Software Engineering
N. Hari Narayanan, Professor of Computer Science and Software Engineering

Abstract

Today, the technology used for voting does not fully address the issues that disabled voters are confronted with during elections. Every voter, regardless of disability, should be able to vote and verify his or her ballot during elections without the assistance of others. In order for this to happen, a universal design [12] should be incorporated into the development of all voting systems. The research presented in this thesis embraces the needs of those who are disabled. The primary objective of this research was to develop a system in which a person, regardless of disability, can efficiently, anonymously, and independently write-in a candidate's name during an election. The method presented here uses speech interaction and name prediction to allow voters to privately spell the name of the candidate they intend to write-in. A study was performed to determine the effectiveness and efficiency of the system. The results of the study showed that spelling a name using the predictive method developed is an effective and efficient solution to the aforementioned issues.

Acknowledgments

First and foremost, I would like to thank my lord and personal savior, Jesus Christ, without whom I would be nothing. I would also like to thank my wonderful husband, Jeremy Dawkins, for his continuous love and support. I would like to thank all of my family and friends who encouraged me to make it this far. I especially want to thank my mother, Francine Wright, for always pushing me to reach for the stars, and my father-in-law, James F. Dawkins, for consistently demanding As. Additional thanks must be given to Dr. Juan Gilbert for his continuous support, encouragement, mentorship, and guidance throughout this entire process. I wish to thank the remaining members of my committee, Dr. Cheryl Seals and Dr. Hari Narayanan, for their reviewing and advising efforts. I want to thank my friends and colleagues who took the time out to review this thesis. To the members of HCCL at AU and other fellow graduate students, I am grateful for your advice, support, and assistance in achieving this goal.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

The 2000 United States Presidential Election will always be remembered for its voting irregularities. The issues with the ballot design during that election led to skepticism of other voting systems and technologies. Not only were there questions regarding the difficulty interpreting the voter's intention, the focus also shifted to the issues surrounding disabled voters. The key issue was that disabled voters needed a way to vote independently and anonymously, while still maintaining system security and efficiency. Every voter, regardless of disability, should be able to vote and verify his or her ballot during elections privately, without assistance. Today, a properly designed interface is one of the key aspects to running a successful election.

As technology for electronic voting systems continues to develop, there is an increased need for universal design in these systems [2]. A universal design ensures that systems are as usable as possible by as many people as possible regardless of age, ability or situation [12]. By focusing on the voter and their needs, the design of electronic voting systems will far surpass the ballot designs of the 2000 election.

With the security of voting systems constantly being a major concern, it is often difficult to implement voting technology that incorporates a secure universal design. Some developers today address this issue through the design of their electronic voting systems [35]; however, these electronic voting systems have yet to integrate universal design into the writing-in of a candidate’s name.

The objective of this research is to develop a system in which a person, regardless of disability, can efficiently, anonymously, and effectively spell a candidate's name through speech interaction. It is necessary to give all voters the ability to write-in a candidate independently, which sparked the notion of a universally accessible approach to write-in a candidate's name. The method presented in this thesis is a predictive approach to spelling through speech interaction. This allows voters to quickly and anonymously spell a candidate's name for any position or office during the voting process. The study performed intends to capture and analyze the effectiveness of writing in a candidate's name anonymously through speech. The results of this study could lead to the adaptation of this system in search functions for various other applications.

Chapter 2

Literature Review

2.1 Election Write-In Compliance

In United States elections, voters have the option to write-in a candidate's name for office. Because election law is mandated by each individual state and not federally, laws pertaining to writing in a candidate vary across all states [16]. Most, but not all states only allow write-in candidates for general elections. Similarly, some states require people to pre-register as a write-in candidate for an election, while others do not. Some states do not allow candidates to be written-in at all [16]. Due to the large variance in election law across states, it is impractical to discuss them in entirety for this thesis. This section will highlight the states of Alabama, Maryland, and California to express the variety amongst all states in general.

The state of Alabama uses paper ballots in their elections rather than electronic voting machines. As such, voters literally have to *write* in a candidate's name on the ballot. Alabama only allows voters to write-in a candidate's name for non-municipal general elections. The general election ballots have a space under each office for the voter to write-in any name not printed on the ballot. In order to vote in this manner, voters must write the desired candidate's name in the space provided on the ballot and register the vote by marking the designated write-in space for that office [4].

Polling places in the state of Maryland use electronic voting machines for elections. Voters can vote using a touchscreen ballot, audio ballot, or provisional ballot. If a voter has a disability that prevents him or her from writing in a candidate's name, an election official enters the voting booth to assist the voter [44]. In Maryland, writing-in a candidate is not allowed in primary elections [26]. If a candidate intends to be elected as a write-in candidate, s/he must file a certificate of candidacy prior to the election [27, 28].

California state election law allows any person to be written in for any public office for any election [11]. For the voting systems in California, the names of write-in candidates can be written on the ballots in the space provided, whether directly beneath the list of candidates or otherwise mentioned in the voting instructions [10]. Some California voting systems do not allow pre-printed stickers, stamps, or other unapproved devices to be used to write-in (or stamp-in) a candidate's name [36]. Like Maryland, people who intend to get elected by means of write-in, need to be certified by filing a statement of write-in candidacy prior to the election [11].

2.2 Universal Accessibility

The following sections are aimed at highlighting accessibility in computing systems. Universal design is first discussed from a general computing perspective, and then discussed as applied to voting, specifically electronic voting systems. Universal Accessibility is the underlying motivation of this research.

2.2.1 Universal Design

Universal design is a key feature that should be incorporated into the design of any computer system. Universal design has been researched by different institutes and organizations [12, 20, 29], and has been defined similarly amongst them. [12] states that universal design is an approach to the design of all products and environments to be as usable as possible by as many people as possible regardless of age, ability, or situation, and that it benefits everyone by accommodating limitations. Accordingly, [29] adds that the intent of universal design is to simplify life for everyone by making products, communications, and the built environment more usable by as many people as possible at little or no extra cost. [20] adds another context by stating that universal design is a framework for the design of places, things, information, communication and policy that is to be usable without special or separate design, and is an orientation of user experience to any design process. Simply stated, universal design is human-centered design of everything with everyone in mind [20].

2.2.2 Universal Accessibility in Voting

As a result of the major issues faced in the 2000 United States Presidential Election, the Help America Vote Act (HAVA) of 2002 was created [17]. HAVA aimed to prevent these problems from happening in future elections. From HAVA, the United States Election Assistance Commission (EAC) was established. One of the goals of the EAC was to adopt Voluntary Voting System Guidelines (VVSG), which expand access for individuals with disabilities to vote privately and independently [40]. The VVSG is the third revision of voting system standards, following the 1990 and 2002 Voting System Standards (VSS). In 2007, the VVSG was made public. The VVSG now addresses the advancement of technology and provides requirements for voting systems to be tested against to ensure functionality, security, and accessibility [34].

It is now necessary for existing and novel electronic voting systems to implement a universal design. Chapter 3 of the 2007 VVSG proposes requirements for the usability and accessibility of electronic voting systems [2]. Due to the diversity amongst people voting in elections, a universal design is essential to the success of electronic voting systems. The VVSG states that all voters must have access to the voting process without discrimination, and that the voting process must be accessible to individuals with disabilities, including non-visual accessibility [40]. It also states that the voting system should be independently accessible to as many voters as possible, which further emphasizes the need for a universal design.

2.3 Word Prediction

Word prediction and word completion are common phrases used to describe suggestion methods for entering text. Many systems are described using the terms word prediction and word completion interchangeably, whereas others define the two as different techniques [42]. For the purposes of this thesis, the approach discussed is identified as word, or name, prediction.

Today, there is a plethora of different methods used for word prediction. Word prediction is often defined as a design in which systems predict the word the user wants to type, based on what s/he has typed so far [38]. Word prediction initially was an assistive solution to enter text for people with motor impairments [22]. Word prediction has since evolved into a way for people to expedite their typing rates for text messaging on mobile devices [9]. This section describes popular methods used today.

2.3.1 T9

T9 (Text on Nine Keys) is a mobile text input system, originally introduced by Tegic Communications, designed to make it possible for users to type as fast as they can think [39]. T9 significantly improved text entry on the phone's fixed, 9-button, keypad because it guesses the word desired from the text that the user has already typed by combining the words from its dictionary with the input it received from the user [23]. Using T9 text input, users are able to type words on a mobile phone using just one key press per letter [30]. On most cellular phones with T9 capabilities, when a user presses a key, the most used letter of that key that fits with the keys already pressed is displayed to the user, and s/he indicates that the word is complete by pressing the “space” key. If the desired word is not displayed initially, the user can simply press the ‘next’ key until the desired word appears. For example, to type the word “biking,” the user would press the sequence, “2-4-5-4-6-4” (for b-i-k-i-n-g). The most common word for that sequence of key presses is “ailing”, which is displayed to the user. When the user presses the “next” key, the word “biking” appears, and the user presses the “space” key to accept.

XT9 Smart Input is an enhancement that goes a step further than T9, by predicting the word the user intends to type [31]. As users type, not only is the T9 method applied to the word, but in addition to that, a complete word is suggested to the user. As applied to the previous example, after the user types the sequence “2-4-5-4”, meaning “a-i-l-i”, the system predicts the word to be spelled and displays “ailing” as a possible choice. The user can accept this word by pressing the “space” key, or deny this word by pressing the “next” key or continuing to type the word.

2.3.2 LetterWise

LetterWise is a word prediction system that uses a probability of letter sequences to guess the next intended letter. Unlike the dictionary based T9 Text Input system, LetterWise takes less memory and allows entry of unconventional words [24]. Rather than store full words in a database, this method only needs to store word prefixes. LetterWise suggests letters using probabilities based on language behaviors. As users type, the system selects the most common letter to display based on the letters already selected and the prefixes in the database.

2.4 Information Security

When it comes to electronic voting systems, information security is a huge issue because of the necessity of voter and ballot privacy. For the system discussed in this thesis, it is most relevant due to the need of a database in the design. Most systems that incorporate a database in the design communicate with the database over a network. Transferring voter information over a network is not feasible in this case because there is the potential for tampering with ballots. The Voter Confidence and Increased Accessibility Act of 2009 [41], bans wireless devices and Internet connections in voting and tabulating machines [25]. For this reason, it is necessary for the system presented to utilize a local database.

2.4.1 Local Databases

SQLite is an open source embedded SQL database engine. It is an in-process library that implements a serverless SQL database engine [3]. One of the good things about SQLite is that it is a local database that has the option of being loaded in memory. It is written in ANSI-C and is compiled by standard C compilers. Being a C programming language-based system is the primary reason for it being disregarded for the design discussed in this thesis, since the system presented here was written in JAVA.

H2 is another type of open source database engine [15]. It uses a JDBC driver, and is JAVA based, which enables it to be incorporated easily into programs written in JAVA. H2 can be run from disk space or in memory, making it a potential candidate for use in secure voting systems. The primary reason this database was not chosen for the predictive system is that it is not robust enough, or as developed as the HSQLDB discussed next.

The HyperSQL Database (HSQLDB) is an open source, SQL relational database engine written in JAVA [19]. HSQLDB also has a JDBC driver, and can be loaded in memory for quick access. It has been tested for stability and reliability, and is the fastest SQL relational database engine available [19]. It has the ability to execute almost every SQL command, including join, count, sum, and max. HSQLDB comes standard with a database GUI tool for database management. For these reasons, this database engine was the optimal choice in databases for the project design for the name database.

Chapter 3

Background

3.1 Election Write-Ins

The method of writing in a candidate's name for a particular United States governing office dates back to the early 19th century [32]. Prior to the 1800s, voters would simply call out their choices to a judge and election clerks tallying the votes [21]. After the 12th amendment was passed in 1804, paper ballots became the standard method for voting. Although paper ballots were the new standard, they were not government issued. For this reason, voters brought their own slips of paper as the ballot, on which they wrote candidate's names [18].

Gradually, candidates began handing out their own ballots, on which their name was already printed. The remainder of the space on these ballots listed the other offices of the election with space available for names to be written-in (Figure 3.1.1) [21]. Paper ballots evolved into fully printed party ballots, listing all candidates of a party for every office (Figure 3.1.1) [18]. Voters could cast a straight ticket by simply dropping the ballot in a locked box. If the voter wanted to vote for a candidate not printed on the ballot, s/he had to cross out the name printed and write the name of his choosing [14].

Today, a write-in candidate is a candidate for public office whose name does not appear on the ballot, but whose name must be written on the ballot by voters [43]. Some districts require write-in candidates to register as official candidates prior to an election.

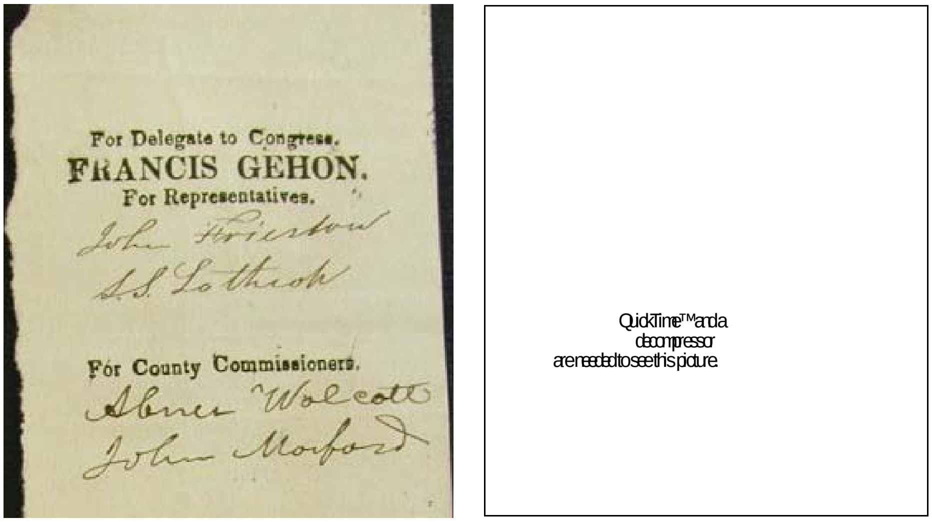


Figure 3.1.1: Example Paper Ballots

Different states and jurisdictions have various rules and regulations for writing in a candidate, especially since many jurisdictions are advancing towards utilizing electronic voting systems. For example, Howard County, Maryland only allows write-in votes in general elections, and provides regulations for writing in a candidate using three different ballots: touchscreen, audio, and provisional [44]. Their ballots have predetermined candidates, and voters have the option to write-in a candidate's name. The laws for writing-in candidates vary by jurisdiction.

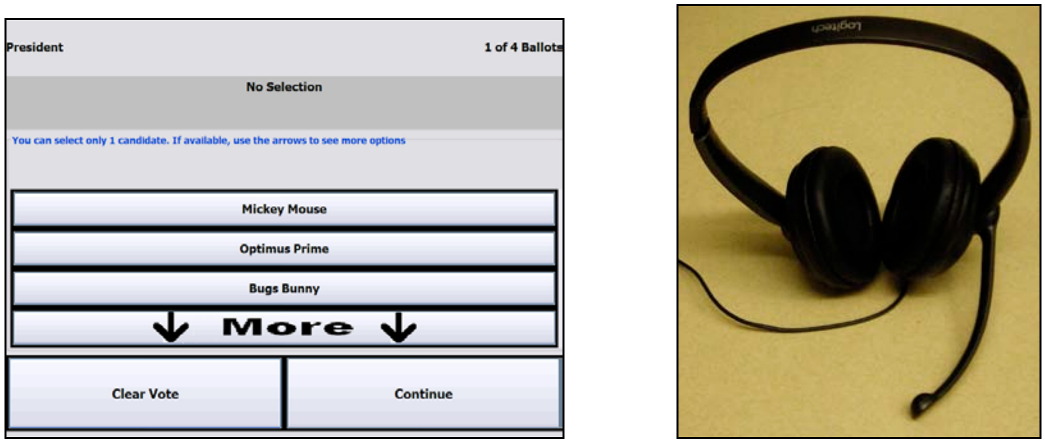
3.2 Prime III

Prime III is a research prototype electronic voting system. It is a secure, multimodal electronic voting system that delivers the necessary system security, integrity and user satisfaction safeguards in a user-friendly interface that accommodates all people regardless of ability [35]. The Prime III system is multimodal in that it uses multiple interaction methods. Voters are able to cast their votes with this system through visual interaction and/or through speech interaction; meaning voters can see and/or hear the candidates' names and other options throughout the voting process.

This multimodal approach to electronic voting enables Prime III to incorporate a universal design, as discussed in Chapter Two, which allows nearly all voters to cast their votes independently and privately. Since Prime III has a universal design, any person who has a visual, cognitive, or motor disability can vote. With Prime III, if a voter has a hearing impairment or disability, s/he can vote using the touchscreen interface (Figure 3.2.1a [35]) to select candidates, navigate the ballot, and to cast and review the final ballot. Conversely, if a voter is visually impaired or disabled, s/he is able to vote using the speech interface (Figure 3.2.1b) by speaking to the system. Through speech, the voter can select candidates, navigate the ballot, and review and cast the final ballot just as sighted voters are able to do. Alternatively, a voter who may have mild visual, speech, and/or motor impairments may choose to vote using the speech and touch interfaces simultaneously.

Due to the anonymous nature of voting systems, the candidates that the voter selects must be kept private. Since Prime III integrates speech interaction into the voting process, bystanders may be afforded the opportunity to compromise the privacy of the voter. This presents an issue in voter - ballot anonymity. Bystanders must not be able to hear whom a voter selects for any office, or a voter's decision for any proposition. Therefore, during the voting process, voters cannot simply say the name of the candidates for which s/he wishes to vote. The speech interface of Prime III implements an interaction in which the voter does not need to explicitly verbalize for which candidate they intend to vote.

The Prime III system uses speech to convey the information on the screen to the voter (e.g. candidates listed for a particular office). Each option is presented to the voter in random order, and the system receives input from the voter through speech. When an option is presented, the voter chooses the option by speaking, "Vote."



(a) (b)

Figure 3.2.1 (a) Touchscreen Interface (b) Speech Interaction Headset

If the voter does not wish to choose the current option, they do not say anything and the system moves on to the next prompt. An example dialogue is as follows:

> Prime III: “To vote for the Democratic Party, say vote <beep>”
> Voter: <says nothing>
> Prime III: “To vote for the Republican Party, say vote <beep>”
> Voter: <says nothing>
> Prime III: “To vote for the Green Party, say vote <beep>”
> Voter: “Vote”

In this example, the voter chose to vote for the green party. With this type of interface, voters make their selections by simply saying “Vote.” Therefore, instead of a voter’s actual choice, bystanders only hear the voter saying “Vote,” which ensures the privacy of the voter and the anonymity of the voter’s ballot.

The universal accessibility and anonymous nature of electronic voting highlights the incompleteness in the design of writing in a candidate’s name with Prime III. Currently, voters have the ability to write-in a candidate’s name in one way: using an onscreen keyboard (Figure 3.2.2). When a voter chooses not to vote for a predetermined candidate and to write-in a candidate’s name, the keyboard is shown, and the user must use the touchscreen to type the candidate’s name.

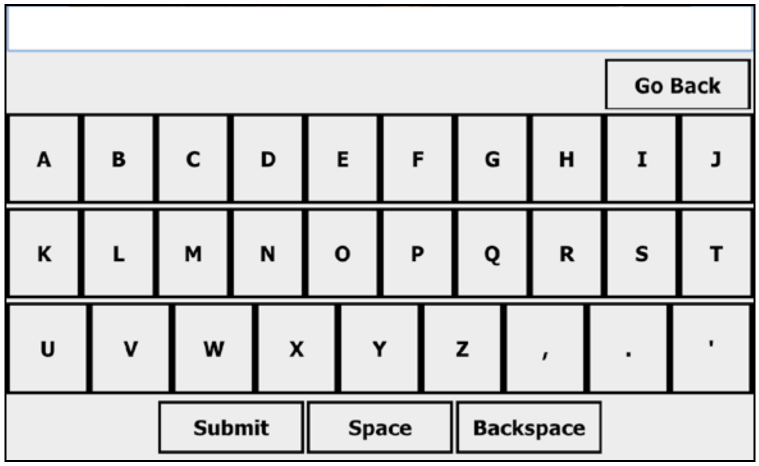


Figure 3.2.2: Prime III On-screen Keyboard

Since this portion of the system is not a multimodal design, the voter must be sighted to write-in a candidate’s name.

Chapter 4

Problem Statement

4.1 Problem Statement

Currently, there is no solution for writing in a candidate's name that is universally accessible. As stated previously, developing systems with a universal design ensures that the system can be used by anyone, regardless of abilities or disabilities. Prime III, like other electronic voting systems today, simply cannot accommodate a range of voters due to its current write-in system through an on screen keyboard. In order for voters with visual or motor impairments to vote, a voting official must enter the voting booth with him or her to write, or type, the candidate on the ballot for which the voter intends to vote. The lack of multimodality and accessibility in these write-in methods only accommodates sighted voters. This violates the privacy of the voter and the anonymity of the voter's ballot.

The most fitting solution to this problem of voter privacy is to utilize a multimodal voting system that incorporates speech interaction. With the addition of speech, voters, regardless of physical disability, have an option to vote independently. In order to write-in a candidate, a voter could simply speak aloud the name of the person who they intend to write-in. The integration of the speech feature alone enables the system to have a universal design. However, this system is not practical. During election peak times, polling places may have a large voter turnout [33]. With the large number of voters at polling places at any given time, privacy is an enormous issue. In accordance with the EAC, the voting process must preserve the secrecy of the ballot. The voting process should preclude anyone else from determining the content of a voter's ballot, without the voter's cooperation. If such a determination is made against the wishes of the voter, then his or her privacy has been violated [2]. If a voter is required to explicitly say the name of the candidate for which they intend to write-in, any bystanders within the polling place may be able to hear that name, and know for whom that person voted, thereby violating the voter's privacy and ballot anonymity.

In order to secure voter privacy through speech interaction, voters must communicate with the system using the speech interaction method of Prime III. As explained in section 3.2, this approach allows a voter to make selections throughout the voting process by simply saying, "vote" in response to the system's prompts. Using this method for writing in a candidate's name has its challenges. The system cannot simply prompt names to the voter until the system gets to the name the voter intends to write-in. There are an infinite number of names the voter would have to choose from. For example, it would not be viable for the dialogue to be as follows:

> Prime III: "To vote for the Bob Smith, say vote <beep>"
> Voter: <says nothing>
> Prime III: "To vote for the Bill Smith, say vote <beep>"
> Voter: <says nothing>
> Prime III: "To vote for the Billy Smith, say vote <beep>"
> Voter: <says nothing>

.
.
.

If the systems simply made uneducated guesses of the desired name, it would be impossible for the voter to write-in a candidate.

A solution to this problem would be for the voter to spell, rather than say, the desired candidate's name. However, due to voter privacy, the voter cannot simply spell a name aloud. Spelling a write-in candidate's name can only be done privately if the Prime III method of getting input data from the voter, through speech, is applied to the design of the system. Using this method, the system would need to prompt the voter to determine the correct letters to spell the desired candidate's name. This would have to be done for the spelling of the entire name. For example, to spell the name, "Bob," the dialogue would be as follows:

> Prime III: "If the first letter of the candidate's name is A, say vote <beep>"
> Voter: <says nothing>
> Prime III: "If the first letter of the candidate's name is B, say vote <beep>"
> Voter: "Vote"
> Prime III: "If the second letter of the candidate's name is A, say vote <beep>"
> Voter: <says nothing>
> Prime III: "If the second letter of the candidate's name is B, say vote <beep>"
> Voter: <says nothing>
> Prime III: "If the second letter of the candidate's name is C, say vote <beep>"
> Voter: <says nothing>
>
> .
> .
>
> Prime III: "If the second letter of the candidate's name is N, say vote <beep>"
> Voter: <says nothing>
> Prime III: "If the second letter of the candidate's name is O, say vote <beep>"
> Voter: "Vote"
>
> .
> .
>
> Prime III: "If the third letter of the candidate's name is B, say vote <beep>"
> Voter: "Vote"

Thus far, this is the best solution. This approach to spelling a candidate's name encompasses voter privacy, integrity, and universal accessibility. However, the above example implements a linear search to spell a write-in candidate's name. For each letter of the candidate's full name, the voter may have to traverse each of the 26 letters of the alphabet. Spelling using this method would take an extremely long time, especially if the letters of the candidate's name were at the end of the alphabet (i.e. "Robert Smith"), or if the candidate's name has several letters (i.e. "Christopher Washington"). Time is a vital factor in voting. Voters want to make their selections and cast their ballots in a reasonable amount of time. This straight linear approach to spell the name of a write-in candidate is long and undesirable, leading to the research discussed in this thesis. The overall objective of this research is to propose a method to write-in a candidate's name that addresses the issues of time, privacy, and accessibility.

Currently, there is no method to spell a name for writing in a candidate that incorporates a universal design and meets the requirements set forth by the EAC; no system allows an individual with visual or motor impairments to spell a candidate's name privately and securely. In order to solve these major issues, a predictive spelling method was created using speech interaction. The hypothesis is that the predictive spelling method through speech interaction will take less time to spell a candidate's name than the aforementioned linear approach.

Chapter 5

Design

5.1 Design Overview

The novel approach for writing in a candidate presented in this thesis is implemented with a universal design that is also private and time effective. The proposed design solution utilizes alphabet clustering and implements name prediction as opposed to the linear search method discussed in the Problem Statement. This solution proves to be more time effective for letter selection, and the overall name selection.

Rather than using linear search to traverse the alphabet, which may take an extensive duration of time to complete, this design breaks down the alphabet into clusters of letters, which are then are presented to the voter. The voter then spells a candidate's name by selecting from these letters and the system performs name prediction similar to the methods used in predictive text technology such as XT9 (discussed in Chapter 2). Like in XT9, the voters spelling with our speech system have the option to select from the suggestions made based on the letters spelled. While XT9 utilizes a dictionary database to predict words that the user may intend to type, this system was developed using a database containing only first and last names that the user may intend to spell.

For each letter of the candidate's name, the clusters are presented to the voter for selection using the method discussed in Chapter Three. The voter begins by making the proper selections to spell the candidate's last name. The system first prompts the voter with the alphabet clusters. Once the voter selects the desired cluster, the system then prompts the voter with the letters contained in that cluster. The voter then chooses a letter, and the system moves on to get the next letter of the desired candidate's name. Following every new letter selection, the first cluster presented for the next letter is a cluster of the three most common letters to follow the letters already chosen.

After the voter selects the first three letters of the candidate's name, the system then suggests three names, one of which the voter may intend to write-in. The names suggested are chosen because they have the highest probability to be written in. If the voter selects one of the names suggested, the process is repeated for the intended candidate's first name, resulting in the chosen candidate's full name being written in for the corresponding office on the ballot. If the voter does not intend to write-in one of the names suggested, s/he continues the process of selecting clusters, then letters, until the correct name is suggested, or the name has been spelled in full (see Table 5.5 for a full example).

5.2 Cluster Selection

The alphabet is broken down into four clusters of five letters, and one cluster of six letters (Table 5.2.1). For the first letter of each of the candidate's names, given name and surname, the voter is prompted to choose from one of the five clusters. For each letter to be spelled after the first letter, there is an additional cluster of 3 letters presented to the voter. This cluster contains the most common next letters, given the letters the candidate has already chosen. For every letter, with the exception of the first letter, the first cluster presented to the voter is the most common letter cluster. This expedites the selection process since the voter is able to make his or her selection at this point, rather than making a selection from the five standard clusters. If the next letter of the name is *not* in the most common letter cluster, the voter is then prompted to select one of the five standard clusters (Table 5.2.1).

Cluster Letters
A, B, C, D, E
F, G, H, I, J
K, L, M, N, O
P, Q, R, S, T
U, V, W, X, Y, Z

Table 5.2.1: Standard Letter Clusters

The standard clusters are generally presented to the voter in alphabetical order. The first of these clusters presented to the voter is chosen at random, with the prompts for the remaining clusters following in alphabetical order, in a round robin fashion. An example of the order in which clusters may be presented is shown in Table 5.2.2. The purpose of this randomization is to secure ballot anonymity by ensuring that bystanders will not be able to piece together for whom the voter voted.

5.3 Letter Selection

Once the voter selects the correct cluster containing the next letter of the desired candidate's name, s/he is prompted to choose amongst those letters. The letters presented by the system are dependent on the cluster the voter selected (see Table 5.5). If the voter selects the cluster of letters {A,B,C,D,E}, s/he is prompted to choose from those letters within that cluster. If the voter selects the cluster of the most common letters, for example, {R, A, E}, s/he is prompted to choose a letter from that common letter cluster. Once the desired letter is chosen, the system moves on to the set of prompts for the voter to select the next letter of the write-in candidate's name (see Table 5.5).

5.4 Name Database

This prediction system for writing in a candidate's name is made possible through the use of a local database of names.

Cluster Type	**Last Name First Letter**	**Last Name Second Letter**
Most Common Letters	-----	R, A, E
Standard	P, Q, R, S, T	F, G, H, I, J
Standard	U, V, W, X, Y, Z	K, L, M, N, O
Standard	A, B, C, D, E	P, Q, R, S, T
Standard	F, G, H, I, J	U, V, W, X, Y, Z
Standard	K, L, M, N, O	A, B, C, D, E

Table 5.2.2: Example Cluster Prompt Order

This database contains the most common names in the United States [8]. Taken from the United States census in 2000, each name was given a category and a rank. The different categories of names are surnames, male given names, and female given names. Within these categories, each name was given a rank based on popularity. The names that were used most frequently are ranked at the top of the list, while the names infrequently used are at the bottom of the list. The database used in this design contains a table of the top 1000 ranked surnames from the 2000 US Census. The database also has a table for given names; containing the top 1000 ranked male names, and the top 1000 ranked female names.

The schema for the name database is shown in Figure 5.4. For example, the name "Jones" is ranked number five in the most common surnames from the 2000 Census. For Jones, the "lastNames" table in the database would have a record where name='Jones' and ranking=5. The "lastNameSpellings" table in the database would have five records for the name "Jones", for each of the letters in the name. The records would be ranking=5, name='Jones', position=0, and letter='J'; ranking=5, name='Jones', position=1, and letter='O'; ranking=5, name='Jones', position=2, and letter='N'; ranking=5, name='Jones', position=3, and letter='E'; ranking=5, name='Jones', position=4, and letter='S'. The first names and surnames in the database are also stored in this fashion.

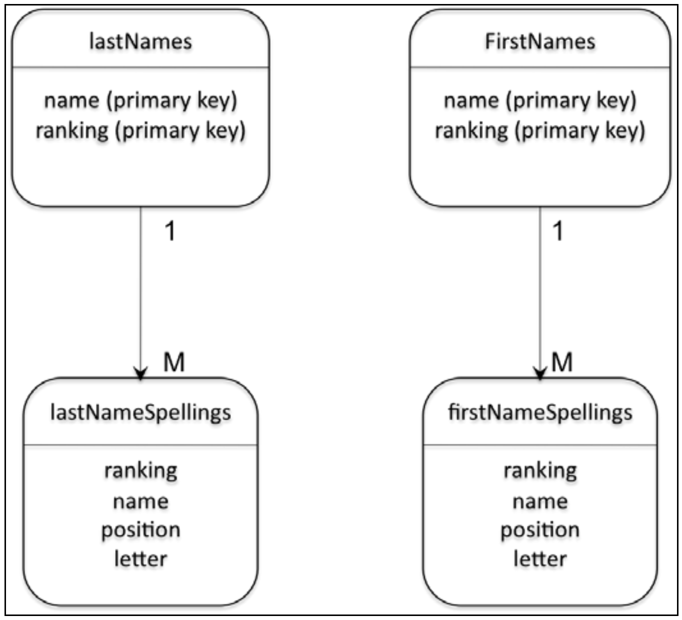


Figure 5.4: Name Database Schematic Diagram

5.5 Name Prediction

In order to effectively reduce the amount of time a voter spends to write-in a candidate's name, this system utilizes a name prediction method built on the name database described in the previous section. Essentially, the predictions are suggestions to the voter of names that s/he may potentially spell. The names suggested are pulled from the name database depending on the letters already chosen by the voter. If one of the predicted names is correct, the voter does not need to go through the entire spelling process.

The name suggestions are strictly based on the clusters and letters chosen by the voter. When a voter selects a cluster, the system can suggest the most common (highest ranked) name that has a first initial as one of the letters in the cluster. For example, if the voter is selecting the first letter of the candidate's last name, and chooses the cluster "F, G, H, I, J," the system suggests "Johnson" to be the candidate's last name. Similarly, when a voter selects a letter, the system can suggest the most common name from the letters selected.

Interaction Mode	Interaction	Letters Already Selected
System	**Say vote if the first letter of the candidate's last name is A, B, C, D, or E**	--
Voter	*Vote*	--
System	**Say vote if the first letter of the candidate's last name is A**	--
Voter	*<says nothing>*	--
System	**Say vote if the first letter of the candidate's last name is B**	--
Voter	*<says nothing>*	--
System	**Say vote if the third letter of the candidate's last name is C**	--
Voter	*Vote*	C
System	**You have selected the letter C. Say vote to delete this letter.**	C
Voter	*<says nothing>*	C
System	**You have selected C as the candidate's last name. Say vote if you are finished spelling the last name.**	C
Voter	*<says nothing>*	C
System	**You will now select the second letter of the candidate's last name.**	C
System	**The next letters are the most common letters. Say vote if the second letter of the candidate's last name is A, E, or O**	C
Voter	*Vote*	C
System	**Say vote if the second letter of the candidate's last name is A**	C
Voter	*Vote*	CA
System	**You have selected the letter A. Say vote to delete this letter.**	CA
Voter	*<says nothing>*	CA
System	**You have selected CA as the candidate's last name. Say vote if you are finished spelling the last name.**	CA
Voter	*<says nothing>*	CA
System	**You will now select the third letter of the candidate's last name**	CA
System	**The next letters are the most common letters. Say vote if the third letter of the candidate's last name is M, R, or T**	CA
Voter	*Vote*	CA
System	**Say vote if the third letter of the**	CA

	candidate's last name is M	
Voter	*<says nothing>*	CA
System	**Say vote if the third letter of the candidate's last name is R**	CA
Voter	*Vote*	CAR
System	**You have selected the letter R. Say vote to delete this letter.**	CAR
Voter	*<says nothing>*	CAR
System	**Say vote if the candidate's last name is Carter**	CAR
Voter	*<says nothing>*	CAR
System	**Say vote if the candidate's last name is Carroll**	CAR
Voter	*<says nothing>*	CAR
System	**Say vote if the candidate's last name is Carpenter**	CAR
Voter	*<says nothing>*	CAR
System	**You will now select the fourth letter of the candidate's last name**	CAR
System	**The next letters are the most common letters. Say vote if the third letter of the candidate's last name is L, P, or S**	CAR
Voter	*Vote*	CAR
System	**Say vote if the candidate's last name is Carlson**	CAR
Voter	*<says nothing>*	CAR
System	**Say vote if the third letter of the candidate's last name is L**	CAR
Voter	*Vote*	CARL
System	**Say vote if the candidate's last name is Carlisle**	CARL
Voter	*Vote*	**CARLISLE**

Figure 5.5: Example Dialogue for Spelling Last Name, "Carlisle"

For example, if the voter is spelling the candidate's last name, and has already selected the letters "J," and "A," the system can suggest "James" as the candidate's last name.

In a best-case scenario, the first name the system suggests would be the name the voter intended to write-in. However, if that is not the case, each suggested name the voter rejects (says nothing) adds unnecessary interaction cycles to the spelling process. For this reason, a different approach was taken to suggest names. Because most names could be suggested correctly given the first three letters, the system waits to suggest names until the voter selects the first three letters. Once the first three letters have been spelled, the system knows if there is a potential match in the database. If there is no match, the system continues to let the voter spell the name intended. If there is a name in the database that starts with the letters that the voter already selected, that name is then suggested to the voter. At this time, the system suggests up to three names for the voter to select from. If after these initial three suggestions the system has not suggested the intended candidate's name, the system prompts the voter to continue to spell the candidate's name. From this point on, the system suggests one name after the voter selects a cluster, and one name after the voter selects a letter. If the voter rejects a name, it is never suggested again, so that the intended name has a chance at being suggested. An example of the system dialogue is shown in Table 5.5.

Chapter 6

Experiment and Analysis

6.1 Introduction

The primary objective of this study was to observe and analyze how people interact with the speech interaction predictive write-in system. The goal of the study is to determine the time it takes a voter to use the write-in system developed. The data from this study will then be analyzed to determine the faster of the two methods discussed in Chapter Four – the predictive approach and the linear approach. It is expected that the predictive system will perform significantly faster when spelling a name than the linear system. Additionally, it is expected that the participants in the study will be able to use the system effectively, meaning they will be able to spell names.

6.2 Experiment Method

6.2.1 Participants

The participants in this study were recruited in a number of ways. The study was advertised through mass email to research labs, word of mouth, and some students were offered extra credit in their courses to participate. The advertisement stated that the students would need 15 minutes to do the study, and how to contact the researcher for more information and to set up a time to participate. All students were accepted to participate in the study, given they were at least 19 years of age. A total number of 40 participants participated in this study, 39 of which were undergraduate students.

6.2.2 Procedures

Upon arrival to the study, participants were first asked to read the information letter that was provided to them [Appendix 1]. The purpose of the information sheet was to inform the participant of his or her rights in participating in this study. If the participant was a student receiving extra credit for participating in the study, his or her name was recorded on a document separate from the data collected in the study. This information was not used to identify participant data, and was relinquished over to the course professor upon the completion of the study. All participants were told not to discuss the experiment with friends and classmates to ensure that all participants had an equal knowledge of the study.

The students then were directed to fill out a pre-questionnaire to obtain their demographic information and prior usage with computing [Appendix 2]. Once the pre-questionnaire was completed, a scenario was given to the students, explaining the study. The scenario was to inform the students about the write-in voting process, and to encourage them to treat the study as if it were an actual election. The students then recorded in writing the name they intended to spell, which could be any first and last name of their choosing, with the exception of their own. Students were seated, and then instructed to put on the microphone head set. It was explained to the student that the speech from the system would be coming from the speakers for observational purposes, and that the headset was strictly for the use of the microphone. During the experiment, data collection methods were used which will be introduced in Section 6.3.1.

6.2.3 Materials

There was a variety of equipment and technology used in this study. A laptop computer, microphone headset, speakers, and timer were needed for this study. The laptop computer was used to run the predictive write-in system software. The laptop used was a MacBook computer running on the Leopard operating system [7]. The software on the MacBook used to run and modify the system code was Eclipse. Eclipse is a free open-source Integrated Development Environment used by software developers [13]. The microphone headset and speakers were plugged into the MacBook for use. The speakers were by Altec Lansing Technologies and the microphone headset was a Logitech USB headset. A simple stopwatch timer was used to capture the times for the study. The experiment results were analyzed using Microsoft Excel.

6.3 Analysis

6.3.1 Data Collection Method

The first method of data collection for the study was done through the pre-questionnaire. Each participant was required to fill out the pre-questionnaire so that demographic information about the participants could be collected. Each pre-questionnaire was given a unique id so that the information could be paired with the information collected during the experiment. Demographic information is needed to determine the type of participants in the study. The demographic information tells us what type of disabilities the participants have, if any. It also indicates if English is the participant's native language, their level of education, and other information.

During the study, information was gathered to analyze the participants' use of the system. A data collection sheet was used to record all information during the study [Appendix 3]. This sheet contained the unique id that corresponds with the pre-questionnaire for each participant, the name each participant chose to write-in, and the time taken to spell that name. Also on the data collection sheet was a space for other observations made throughout the study. The results of both the pre-questionnaire and the data collection sheet are presented in the next section.

6.3.2 Results

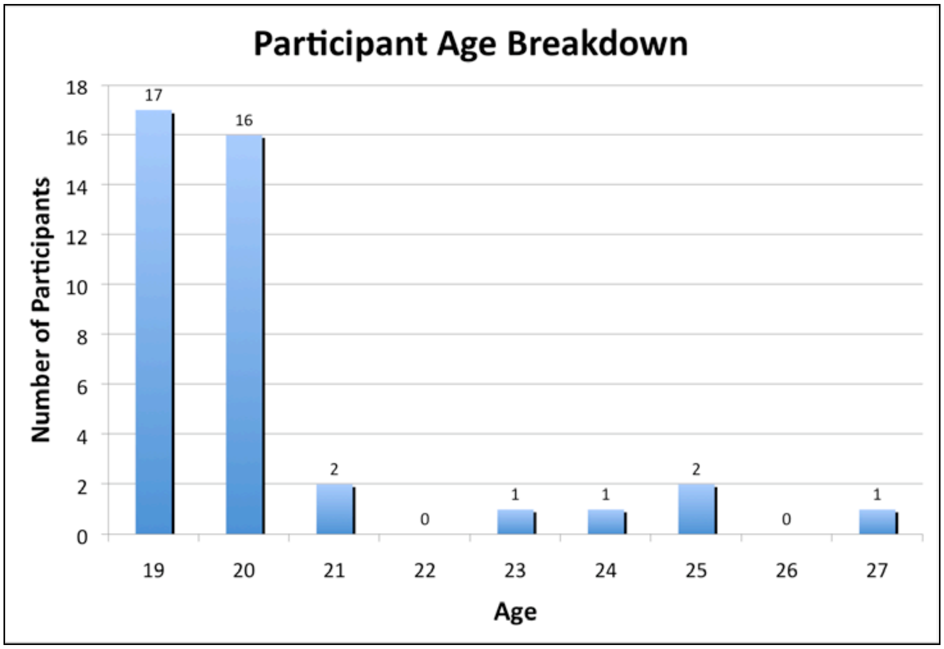
This section presents the quantitative data from the pre-questionnaire and the data collection sheet. Also presented, are calculated comparisons between the predictive write-in method versus the linear search approach.

Participant Results

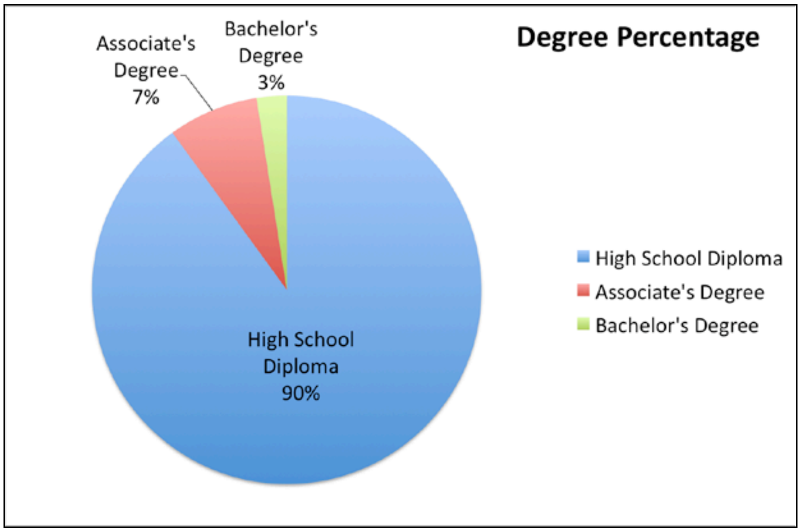
There were a total of 40 participants to participate in this study. Of these participants, 17 were 19 years of age, 16 were 20 years of age, and 6 were 21 years of age or over (Figure 6.3.1a). The age range for the participants was 19 to 27, with an average 20.2 years of age. Given the age of the participants, it is not unusual that 90% of the participants only had high school degrees, while only 7% and 3% had associate's degrees and bachelor's degrees, respectively. As shown in Table 6.3.1, there were 34 males to participate in this study, making up 85% of the participants. 100% of the participants were United States citizens, 95% of which were white. Three of the participants listed that they had disabilities; one said he had dyslexia, another said she was deaf in one ear, and another said he had poor vision.

Predictive Write-in Results

For the study, participants were required to provide a name to spell so that there was no bias amongst the names spelled [Appendix 4]. The average length of the full names chosen was 10.43 letters, with a standard deviation of 2.22. The shortest full name was 7 letters in length, and the longest full name was 16 letters in length. Of the 80 first and last names chosen, 71.3% of the names were in the database and suggested to the user. The average time it took for a participant to spell a candidate's full name was 9.52 minutes, with a standard deviation of 3.83. The median time was 8.42 minutes.



(a) Participant Ages



(b) Participant Degrees Obtained

Figure 6.3.1: Participant Demographic Results

	Number of Participants	**Percentage of Participants**
Male	34	85%
Female	6	15%
Total	40	100%

Table 6.3.1: Participant Gender Results

The average time, for the names given, per letter was 1.09 minutes, with a standard deviation of 45 seconds. These results are shown in Table 6.3.2. The time it took each participant to spell a name is shown in Appendix 4.

Figure 6.3.2 shows a breakdown of times based on the number of letters in the full name spelled. This figure shows the average times taken by participants to spell names of various lengths for the predictive method. Removing the outliers of this chart, the average full name was between 8 and 16 letters, and took an average of 9.23 minutes. These results show that in practice, this system takes much longer than anticipated (see Comparison). Additional observations from the study showed that participant errors were the primary reason that the actual times were much different than what was calculated for the best-case times to spell the same names. The results for these best-case times, along with the best-case times for the linear search approach, are presented in the next section.

Comparison

We calculated, at best case, how long it should take someone to spell the names from the study for both systems [Appendix 4]. We used this method because we expected the linear approach combined with the predictive approach, coupled with a with-in subject study would take too long per participant. In order to determine how long it would take to spell a name, each interaction cycle for the system was broken down and timed. For each method, the sequence of prompts presented to the voter to spell a name is different [Appendix 5]. The sequences were determined for each system, and compiled for each name spelled. The sequences for the predictive write-in method was constructed under the assumption that the names to be spelled are in the system's name database.

	Time to spell full name (minutes)	Number of letters per full name	Average time per letter (minutes)
Average	9.52	10.43	1.09
Standard Deviation	3.83	2.22	.45
Median	8.42	10.00	1.04

Table 6.3.2: Predictive Write-in Statistics

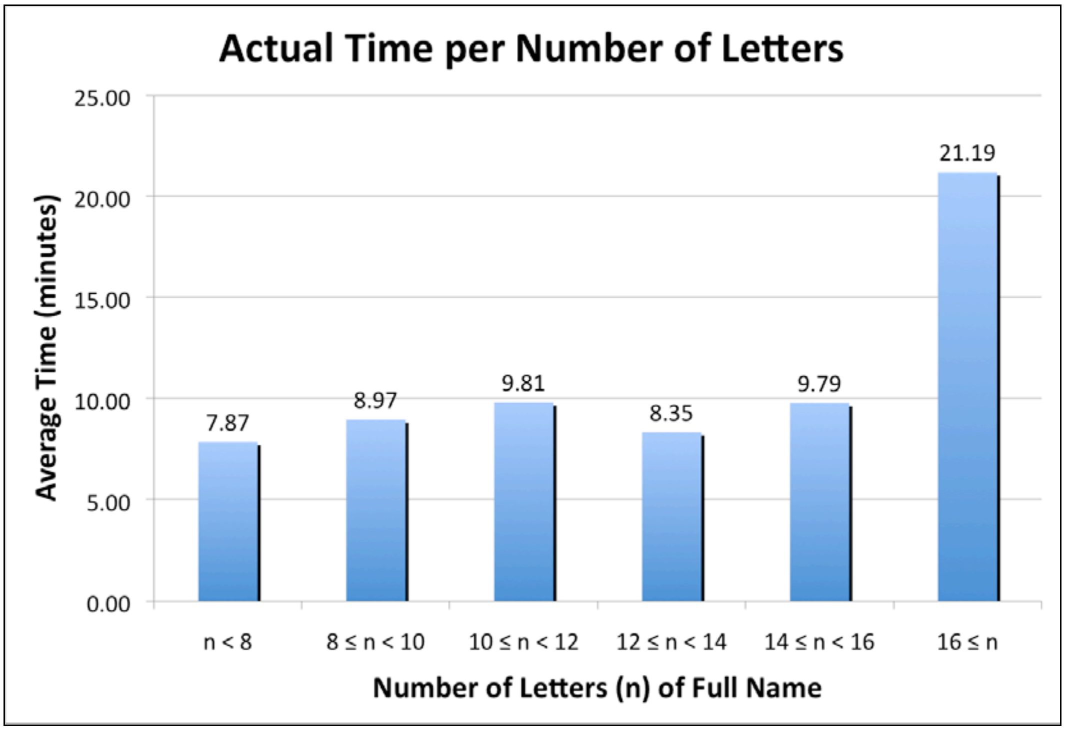


Figure 6.3.2: Average Time to Spell Full Names

The resulting times for each name for both systems are listed in Appendix 4. Figure 6.3.3 shows a breakdown of times for both methods based on the number of letters of the full names that were spelled. This figure shows the average times taken to spell names of various lengths for the predictive and linear methods. The average time for the full names provided in the study for the calculated linear search method was 15.09 minutes, with a standard deviation of 3.86 (Table 6.3.3). The average time to spell the full names for the calculated predictive method was 4.33 minutes, with a standard deviation of 0.17.

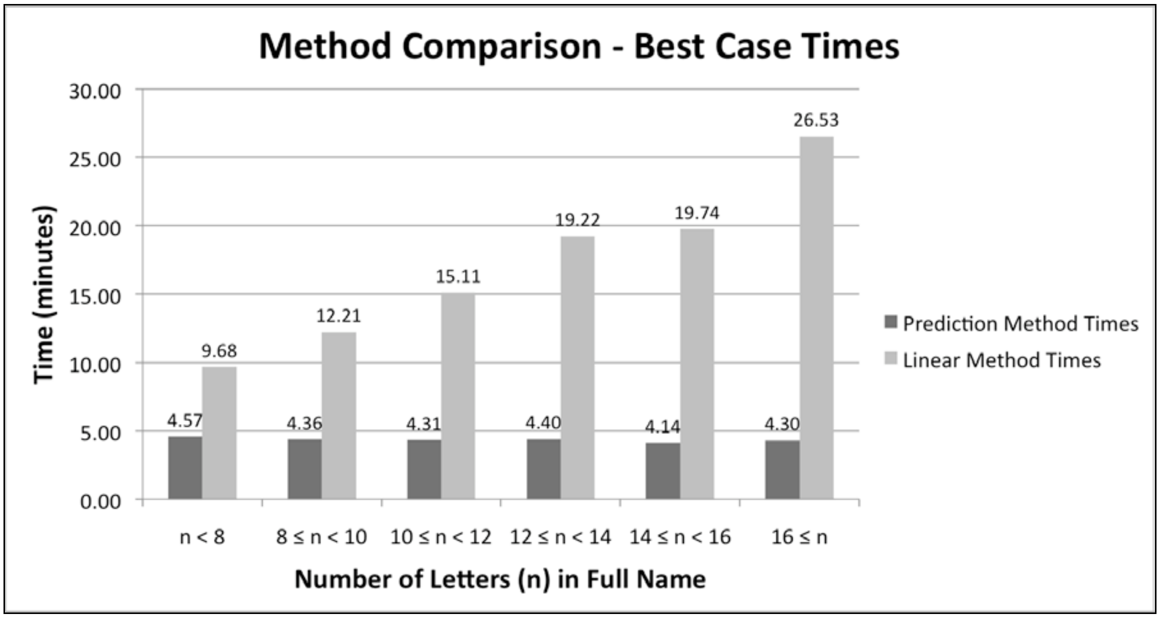


Figure 6.3.3: Method Comparison of Times to Spell Full Names

	Time to spell full name - Predictive Method (minutes)	Time to spell full name - Linear Spelling Method (minutes)
Average	4.33	15.09
Standard Deviation	0.17	3.86
Median	4.34	14.73

Table 6.3.3: Calculated Predictive and Linear Spelling Statistics

The median times for the calculated predictive and linear methods were 4.34 and 14.73, respectively. From these results, we can conclude that, on average, the predictive spelling approach is more than three times faster than the linear spelling approach. The predictive spelling method was effective in that 100% of the participants were able to complete the spelling of the intended names.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The ultimate goal of electronic voting systems today should be to allow anyone to vote privately and independently using a single design. The VVSG (discussed in Chapter Two) provides useful and necessary guidelines to ensure that all eligible citizens have the same access when voting, regardless of a person's disabilities. The primary objective of this research was to embrace these guidelines by developing a system in which a person, regardless of disability, can efficiently, anonymously, and independently write-in a candidate's name during an election. The method designed allows voters to spell a candidate's name discretely through speech interaction. This method uses a predictive approach in order for the voter to get through the voting process of spelling a candidate's name quickly and accurately.

The study performed was designed to test the hypothesis, which states that the method designed for predictive spelling through speech interaction will take much less time to spell a candidate's name than the method of linear search. The study captured the efficiency of writing-in a candidate's name through speech. The results of the study suggest that the predictive approach to write-in a candidate's name was more efficient than the linear spelling approach. However, it was determined that, in practice, the participants took longer than calculated to spell a name using the prediction method. Solutions to this problem and other applications for this design are described in the next section.

7.2 Future Work

The results of the comparison between spelling with linear search and spelling with prediction proved the prediction method to be the method of choice for writing-in a candidate's name. However, in practice, participants took much longer than anticipated to spell a name. From observing the participants throughout the study, it was considered that the number of errors made during the spelling process might have been the primary reason for the time being so long. Future versions of this system will include increased efficiency for error correction.

Because of the similarities amongst the group of participants in the study, results may have been skewed one way or another. It may be beneficial for future studies to include participants of a more diverse demographic. It may also be useful to collect other metrics for determining efficiency, such as, letters required to spell a name, and number of errors made while spelling and where said errors occurred. Recording the interaction during the spelling process may also prove to be a helpful addition to the data analysis.

As this method is further developed, it can be adapted by certain search functions. Search applications that utilize a fixed directory will benefit greatly by using the prediction method discussed. This could be especially helpful for people directories, building directories, or telephony systems.

References

[1] 2005 Voluntary Voting System Guidelines (2005) [online]. http://www.eac.gov/voting%20systems/voluntary-voting-guidelines/2005-vvsg.

[2] 2007 VVSG Chapter 3: Usability, Accessibility, and Privacy Requirements (2007) [online]. http://www.eac.gov/vvsg/part1/chapter03.php/.

[3] About SQLite. http://www.sqlite.org/about.html.

[4] Alabama Code (1975) [online]. http://alisondb.legislature.state.al.us/acas/CodeOfAlabama/1975/coatoc.htm.

[5] Alabama Election Code 17-6-27 (1975) [online]. http://alisondb.legislature.state.al.us/acas/CodeOfAlabama/1975/17-6-27.htm.

[6] Alabama Election Code 17-6-28 (1975) [online]. http://alisondb.legislature.state.al.us/acas/CodeOfAlabama/1975/17-6-28.htm.

[7] Apple MacBook. http://www.apple.com/macbook/.

[8] Butler, R., (2005). http://names.mongabay.com/.

[9] Butts, L. and Cockburn, A., (2001). An Evaluation of Mobile Phone Text Input Methods, Third Australasian User Interfaces Conference, Melbourne, Australia, Conferences in Research and Practice in Information Technology, Vol. 7, pp.55-59. [online] http://www.acs.org.au/documents/public/crpit/CRPITV7Butts.pdf.

[10] California Code (2009) 15340-15342 [online]. http://www.leginfo.ca.gov/cgi-bin/displaycode?section=elec&group=15001-16000&file=15340-15342.

[11] California Code (2009) 8600-8605 [online]. http://www.leginfo.ca.gov/cgi-bin/displaycode?section=elec&group=08001-09000&file=8600-8605.

[12] Center for Universal Design, N.C. State University (2004). Universal Design Education Online Web Site http://www.udeducation.org/learn/index.asp.

[13] Eclipse. http://www.eclipse.org/.

[14] Gill, K., An Illustrated History of Voting Methods and Systems. http://uspolitics.about.com/od/elections/ig/History-of-Voting-Systems/.

[15] H2 Database Engine. http://www.h2database.com/html/main.html.

[16] Helm, B., (2004). Ins and Outs of Write-Ins [online]. http://www.businessweek.com/bwdaily/dnflash/nov2004/nf2004112_5680_db038.htm.

[17] Help America Vote Act. (2002). Public Law 107-252, 107th Congress, United States [online]. http://www.fec.gov/hava/law_ext.txt.

[18] History of the Paper Ballot [online]. http://www.fairvote.org/righttovote/pballot.pdf.

[19] HSQL Database Engine. http://hsqldb.org.

[20] Institute for Human Centered Design (2008). http://www.adaptiveenvironments.org/index.php?option=Content&Itemid=3.

[21] Jones, D., (2003). A Brief Illustrated History of Voting [online]. http://www.cs.uiowa.edu/~jones/voting/pictures/.

[22] Klund, J., Novak, M. (2001). If Word Prediction Can Help, Which Program Do You Choose? [online]. http://trace.wisc.edu/docs/wordprediction2001/.

[23] Learn to Use T9 (2007). http://www.t9.com/us/learn/.

[24] MacKenzie, I.S., Kober, H., Smith, D., Jones, T., Skepner, E., (2001). LetterWise: Prefix-Based Disambiguation for Mobile Text Input, Proceedings of the 14th annual ACM symposium on User interface software and technology, November 11-14, 2001, Orlando, Florida [online]. http://www.eatoni.com/research/lw-mt.pdf.

[25] Making Sure Votes Count and Are Counted (2009). http://holt.house.gov/voting.shtml.

[26] Maryland Election Code 8-205 (2009) [online]. http://www.michie.com/maryland/lpext.dll/mdcode/d676/d8fe/d910/d925?f=templates&fn=document-frame.htm&2.0#JD_el8-205.

[27] Maryland Election Code 5-303 (2009) [online]. http://www.michie.com/maryland/lpext.dll/mdcode/d676/d7a6/d7c8/d7d3?f=templates&fn=document-frame.htm&2.0#JD_el5-303.

[28] Maryland Election Code 5-704 (2009) [online]. http://www.michie.com/maryland/lpext.dll/mdcode/d676/d7a6/d80d/d822?f=templates&fn=document-frame.htm&2.0#JD_el5-704.

[29] North Carolina State University Center for Universal Design (2008). http://www.design.ncsu.edu/cud/about_ud/about_ud.htm.

[30] Nuance – T9 Text Input (2009). http://www.nuance.com/t9/textinput/.

[31] Nuance – XT9 Smart Input (2009). http://www.nuance.com/t9/xt9/.

[32] Official Election Site of San Mateo County (2007). http://www.shapethefuture.org/press/2007/090607.asp.

[33] Polling Place and Vote Center Management (2009) [online]. http://www.eac.gov/election/quick-start-management-guides/docs/09-polling-place-and-vote-center-management.pdf/attachment_download/file.

[34] Press Release: 2005 VVSG Adopted (2005). http://www.eac.gov/voting%20systems/docs/eac-adopts-2005-voluntary-voting-system-guidelines.pdf/attachment_download/file.

[35] Prime III: One Machine, One Vote for Everyone. http://primevotingsystem.org/.

[36] Summary of Qualifications and Requirements for Write-In Candidates (2003). http://www.sos.ca.gov/elections/gov-qual_write-in.pdf.

[37] Technical Guidelines Development Committee (2007). Voluntary Voting System Guidelines Recommendations to the Election Assistance Commission [online] http://www.eac.gov/files/vvsg/Final-TGDC-VVSG-08312007.pdf.

[38] Trnka, K., McCaw, J., Yarrington, D., McCoy, K., and Pennington, C. 2009. User Interaction with Word Prediction: The Effects of Prediction Quality. ACM Trans. Access. Comput. 1, 3 (Feb. 2009), 1-34. [online]. http://doi.acm.org/10.1145/1497302.1497307

[39] Valid Concept (2009). T9: Text on Nine Keys [online]. http://www.validconcept.com/articles-t9.html.

[40] Voluntary Voting System Guidelines Recommendations to the Election Assistance Commission (2007) [online]. http://www.eac.gov/files/vvsg/Final-TGDC-VVSG-08312007.pdf.

[41] Voter Confidence and Increased Accessibility Act of 2009 (2009). Public Law H.R. 2894. [online] http://thomas.loc.gov/cgi-bin/query/z?c111:H.R.2894:.

[42] What is word prediction versus word completion? (2009). http://www.aurora-systems.com/pages/faqpred.html#WPVSWC.

[43] Write-in Candidate. http://dictionary.reference.com/browse/write-in+candidate.

[44] Write In Voting Judges Manual (2008) [online]. http://www.co.ho.md.us/BOE/BOEdocs/Howard_Election_Judges_Manual_2008_Ch05_Approved.pdf.

Appendix 1

Information Letter

AUBURN UNIVERSITY

SAMUEL GINN COLLEGE OF ENGINEERING

(NOTE: DO NOT AGREE TO PARTICIPATE UNLESS AN IRB APPROVAL STAMP WITH CURRENT DATES HAS BEEN APPLIED TO THIS DOCUMENT.)

INFORMATION LETTER
for a Research Study entitled
"A Word Prediction approach to Spelling and Search Through Speech Interaction"

You are invited to participate in a research study to analyze different methods for writing in a candidate's name, i.e. via speech or via touchscreen, to determine if the speech interaction method is comparable to the touch interaction method. The results of this project will be used in presentations, publications, and thesis research. The study is being conducted by Shaneé Dawkins, PhD student, under the direction of Dr. Cheryl Seals, in the Auburn University Department of Computer Science and Software Engineering. You were selected as a possible participant because you are age 19 or older.

What will be involved if you participate? If you decide to participate in this research study, you will be asked to fill out a pre-questionnaire. The purpose of the pre questionnaire is to obtain demographic information and prior usage with computing information. You will spell a full name (first and last) of your choosing twice, with a touchscreen interface and a speech interface. For the touchscreen interface, you will type the name, and submit it to the system. For the speech interface, you will spell the name using the system's program, and submit the name upon completion. Your total time commitment will be approximately 25 minutes.

Are there any risks or discomforts? There are no foreseeable risks.

Are there any benefits to yourself or others? As electronic voting becomes more mainstream, these interfaces will provide a solution to the necessary universal design requirements of voting systems and other applications. We / I cannot promise you that you will receive any or all of the benefits described.

Will you receive compensation for participating? If you are a COMP 1200 student, to thank you for your time, you will be offered 2 extra credit points towards your midterm grade.

Are there any costs? There are no costs associated with participating.

If you change your mind about participating, you can withdraw at any time during the study. Your participation is completely voluntary. If you choose to withdraw, your data can be withdrawn as long as it is identifiable. Your decision about whether or not to participate or to stop participating will not jeopardize your future relations with Auburn University, the Department of Computer Science and Software Engineering.

Department of Computer Science and Software Engineering
3101 Shelby Center for Engineering Technology, Auburn, AL 36849-5347, USA · 334-844-4330 · Fax 334-844-6329 www.eng.auburn.edu/csse/

Any data obtained in connection with this study will remain anonymous. Data will be collected through observations and a questionnaire. Names and other identifiers will not be collected or used. Information collected through your participation may be used to fulfill an educational requirement, published in a professional journal, and/or presented at a professional meeting.

If you have questions about this study, please ask them now or contact Shaneé Dawkins at (334) 844-7001 or e-mail at stw0004@auburn.edu, or Dr. Cheryl Seals at (334) 844-6319 or e-mail at sealscd@auburn.edu.

If you have questions about your rights as a research participant, you may contact the Auburn University Office of Research Compliance or the Institutional Review Board by phone (334)-844-5966 or e-mail at hsubjec@auburn.edu or IRBChair@auburn.edu.

HAVING READ THE INFORMATION PROVIDED, YOU MUST DECIDE IF YOU WANT TO PARTICIPATE IN THIS RESEARCH PROJECT. IF YOU DECIDE TO PARTICIPATE, THE DATA YOU PROVIDE WILL SERVE AS YOUR AGREEMENT TO DO SO. THIS LETTER IS YOURS TO KEEP.

____

Investigator's signature Date

____

Print Name

____

Co-Investigator's signature Date

____

Print Name

Page 2 of 2

Appendix 2

Pre-Questionnaire

<u>Pre-Questionnaire</u>

ID: 1

Age: ________

Gender: ☐ Male ☐ Female

Race / Ethnicity:

☐ Black ☐ White ☐ Asian ☐ Native American ☐ Hispanic ☐ Other ________________

Citizenship: __________________________

Highest Degree Obtained (High School, Bachelor's, Master's, Doctorate, etc): ____________________

Do you have any disabilities? ☐ Yes ☐ No

If yes, please explain: ____

Is English your native language? ☐ Yes ☐ No

Page 16

Appendix 3

Data Collection Sheet

Data Collection Sheet			
Generated ID	Name	Time	Comments

Page 1

Appendix 4

Study Results

ID	Name	Actual Times – Prediction Method: Actual Time	Actual Times – Prediction Method: # Letters	Actual Times – Prediction Method: Average Time per Letter	Calculated Times – Prediction Method: Time Calcuated - First Name	Calculated Times – Prediction Method: Time Calcuated - Last Name	Calculated Times – Prediction Method: Calculated Best Case Time	Calculated Times – Linear Method: Time Calcuated - First Name	Calculated Times – Linear Method: Time Calcuated - Last Name	Calculated Times – Linear Method: Calculated Best Case Time
6	Ciaphas Cain	12.18	11	1.11	2.08	2.08	4.28	8.03	4.13	12.28
7	Brittany Abrams	13.53	14	0.97	1.98	1.89	3.99	13.40	7.41	20.93
8	Jack Jones	11.10	9	1.23	2.27	2.27	4.67	3.93	7.95	12.00
9	Bob Jones	6.28	8	0.79	1.98	2.27	4.38	3.02	7.95	11.08
10	Tony Green	7.06	9	0.78	2.18	1.98	4.28	8.77	6.59	15.48
11	Reese Witherspoon	21.19	16	1.32	2.08	2.10	4.30	6.88	19.53	26.53
12	Lori Woodfin	11.45	11	1.04	1.98	2.10	4.20	6.74	10.84	17.70
13	Bob Billy	11.34	8	1.42	1.98	1.98	4.09	3.02	7.66	10.79
14	Stephen Wilson	4.42	13	0.34	2.18	2.10	4.40	10.94	11.09	22.14
15	Roger Davison	13.48	12	1.12	2.08	2.18	4.38	7.95	10.65	18.71
17	Bob Barker	10.05	9	1.12	1.98	1.98	4.09	3.02	7.04	10.18
18	Bill James	6.32	9	0.70	1.98	2.27	4.38	4.90	6.49	11.51
19	Mickey Mouse	18.16	11	1.65	2.08	2.08	4.28	8.57	8.91	17.60
20	Joe Smith	10.58	8	1.32	2.27	2.18	4.57	4.08	8.53	12.73
21	Jonathan Dunn	11.29	12	0.94	2.27	2.18	4.57	12.14	6.64	18.91
22	James Harmon	7.50	11	0.68	2.27	2.08	4.47	6.49	8.86	15.47
23	Daffy Duck	14.25	9	1.58	2.18	2.18	4.47	5.91	5.29	11.32
24	Daniel Fisher	7.29	12	0.61	2.18	1.89	4.18	6.54	8.47	15.13
25	Guitar Bob	12.26	9	1.36	1.98	1.98	4.09	9.54	3.02	12.68
26	Charles Griffin	7.18	14	0.51	2.08	1.98	4.18	8.90	9.19	18.22
27	Super Man	8.45	8	1.06	2.18	2.08	4.38	9.49	3.89	13.50
28	Daniel John	6.37	10	1.22	2.18	2.27	4.57	6.54	6.06	12.72
29	George Bush	9.47	10	1.35	1.98	1.98	4.09	7.70	6.35	14.17
30	Davis Darwin	11.21	11	1.01	2.18	2.18	4.47	7.17	8.86	16.15
31	Sarah Smith	6.11	10	0.63	2.18	2.18	4.47	6.40	8.53	15.04
32	Kathleen Dulaney	13.16	15	0.47	1.89	2.18	4.18	10.20	10.45	20.78
33	Lane Otto	8.25	8	2.65	1.98	2.27	4.38	4.42	8.29	12.83
34	John Doe	7.35	7	1.64	2.27	2.18	4.57	6.06	3.50	9.68
35	Barack Obama	18.21	11	1.03	1.98	2.27	4.38	5.66	4.94	10.73
36	Charles Haro	9.43	11	0.40	2.08	2.08	4.28	8.90	5.58	14.60
37	Mickey Mouse	8.53	11	1.23	2.08	2.08	4.28	8.57	8.91	17.60
38	Kali Wilson	8.29	10	1.01	1.89	2.10	4.11	4.71	11.09	15.91
39	John Doe	8.38	7	0.90	2.27	2.18	4.57	6.06	3.50	9.68
41	Rob Herring	5.09	10	1.82	2.08	2.08	4.28	4.57	10.16	14.85
42	Taylor Rogers	5.27	12	0.88	2.27	2.08	4.47	10.99	10.12	21.23
43	Wayne Kelley	5.40	11	1.03	2.10	1.89	4.11	7.64	8.96	16.71
44	Bob Jones	6.12	8	0.94	1.98	2.27	4.38	3.02	7.95	11.08
45	Bob Smith	6.31	8	1.78	1.98	2.18	4.28	3.02	8.53	11.66
46	Gabriel McKenzie	5.27	15	0.49	1.98	2.08	4.18	7.74	11.17	19.03
47	John Jones	7.23	9	1.36	2.27	2.27	4.67	6.06	7.95	14.13
	Average Time	9.52	10.43	1.09	2.10	2.12	4.33	6.94	8.03	15.09
	Standard Deviation	3.83	2.22	0.45	0.12	0.11	0.17	2.58	2.88	3.86
	Median	8.42	10.00	1.04	2.08	2.10	4.34	6.64	8.12	14.73

Appendix 5

Prompt Sequences

Prompt Number	**Prompt**	**Prompt Type**	**Interaction Cycle Time (seconds)**
1	You will now be prompted to spell the candidate's {first, last} name.	Which Name	3.90
2	You will now select the {first, second, etc.} letter of the candidate's {first,last} name.	Which Position	3.77
3	The next letters are the most common letters. Say vote if the {first, last} letter of the candidate's {first, last} name is {AEO, CAE, etc}.	Most Common Letter Cluster	8.75
4	Say vote if the {first, last} letter of the candidate's {first, last} name is {ABCDE, FGHIJ, KLMNO, PQRST}.	5 Letter Cluster	6.40
5	Say vote if the {first, last} letter of the candidate's {first, last} name is {UVWXYZ}.	6 Letter Cluster	7.51
6	Say vote if the {first, last} letter of the candidate's {first, last} name is {A,B,C,D,E, etc}.	Individual Letter	5.81
7	Say vote to delete this letter.	Delete	6.00
8	You have selected {letter} as the candidate's {first, last} name. Say vote if you are finished spelling the candidate's {first, last} name.	Check if Finished After 1 Letter	10.22
9	You have selected {letter letter} as the candidate's {first, last} name. Say vote if you are finished spelling the candidate's {first, last} name.	Check if Finished After 2 Letter	10.33

10	You have selected {letters} as the candidate's {first, last} name. Say vote if you are finished spelling the candidate's {first, last} name.	Check if Finished After Letters already spelled (>2)	10.24
11	Say vote if the candidate's {first, last} name is {name}	Suggestion	5.72
12	You have chosen the name {name}. Say vote if this is incorrect.	Name Confirmation	6.68
13	You have chosen the name {first name, last name}. Say vote if this is incorrect.	Full Name Confirmation	7.19
14	You have already selected the letters {letters}	Reminder	2.62 + (.24 for every letter already selected)

Method	**Sequence**
Prediction Method Best Case	[1-2-(4 or 5)-6-7-8-2-3-6-7-9-2-14-3-14-6-7-11-12]* 13 *repeat for first name
Linear Method	{1-(2-6)*-7-8-(2-6)*-7-9-(2-14-6)*-7-10-[(2-14-6)*-7-10]** 12}*** 13 *repeat until prompt for desired letter **repeat until all letters selected ***repeat for first name