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

Shaneé Dawkins
Auburn University
3101 Shelby Center for Technology
Auburn University, AL 36849, USA
stw0004@auburn.edu

Juan E. Gilbert
Clemson University
100 McAdams Hall
Clemson, SC 29634, USA
juan@clemson.edu

Abstract

Today, the technology used for voting does not fully address the issues that disabled voters are confronted with during elections. Voters, including those with most disabilities, 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 should be incorporated into the development of all voting systems. The research presented here embraces the needs of those who are disabled. The primary objective of this research was to develop a system in which a person, 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.

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. All voters, including those with most disabilities, 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 (VVSG Chapter 3, 2007). A universal design ensures that systems are as usable as possible by as many people as possible regardless of age, ability or situation (Center for Universal Design, 2004). 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

Proceedings of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies, pages 44–52,
Los Angeles, California, June 2010.

address this issue through the design of their electronic voting systems (Prime III, 2009); 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, including those with most disabilities, can efficiently, anonymously, and effectively spell a candidate's name through speech interaction. The method presented in this paper 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.

2 Background

2.1 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 (Official Election Site, 2007). Prior to the 1800s, voters would simply call out their choices to a judge and election clerks tallying the votes (Jones, 2003). After the 12th amendment was passed in 1804, paper ballots became the standard method for voting. Voters would bring their own slips of paper as the ballot, on which they wrote candidate's names (History of the Paper Ballot, 2009). Today, a write-in candidate is a candidate whose name does not appear on the ballot. Voters can vote for a write-in candidate by marking the write-in indicator, and writing the candidate's name in space provided on the ballot (Write-in Candidate Requirements, 2010).

2.2 Prime III Electronic Voting System

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 (Prime III, 2009). With Prime III, voters are able to cast their votes through visual interaction and/or through speech interaction. This multimodal approach to electronic voting enables Prime III to incorporate a universal design, which allows nearly all voters to cast their votes independently and privately.

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. Bystanders must not be able to hear whom a voter selects for any office, or a voter's decision for any proposition in order to ensure voter – ballot anonymity. 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) through the use of a microphone headset. When an option is presented, the voter chooses the option by speaking, "Vote" into the microphone. 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: "Vote"

In this example, the voter chose to vote for the Republican Party. Bystanders only hear the voter saying "Vote," instead of a voter's actual choice, 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 1). 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. Since this portion of the system is not a multimodal design, the voter must be sighted to write-in a candidate's name.

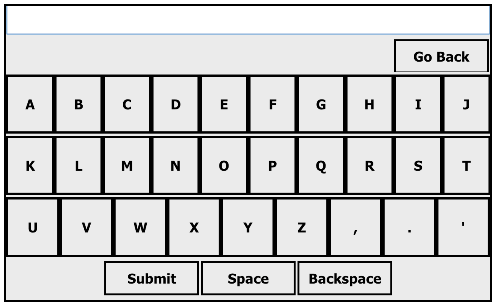

Figure 1. Prime III On-screen Keyboard

2.3 Universal Accessibility in Voting

Help America Vote Act (HAVA) of 2002, was created to prevent the major issues faced in the 2000 United States Presidential Election from happening in future elections (HAVA, 2002). 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 (VVSG, 2007). The VVSG now addresses the advancement of technology and provides requirements for voting systems to be tested against to ensure functionality, security, and accessibility (VVSG, 2005). Chapter 3 of the 2007 VVSG proposes requirements for the usability and accessibility of electronic voting systems (VVSG Chapter 3, 2007). 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 (VVSG, 2007). It also states that voting systems should be independently accessible to as many voters as possible, which further emphasizes the need for a universal design.

3 Motivation

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 most physical disabilities, 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 (Polling Place and Vote Center Management, 2009). With the large number of voters at polling places at any given time, privacy is an enormous issue. In accordance with the Election Assistance Commission (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 (VVSG Chapter 3, 2007). 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 2.2, 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 Doe, say vote [beep]"
> Voter: <says nothing>
> Prime III: "To vote for the Bill Doe, say vote [beep]"
> Voter: <says nothing>

Prime III: “To vote for the Billy Doe, 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 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 presented in this paper. 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.

4 Design

4.1 Design Overview

The novel approach for writing in a candidate presented in this paper is implemented with a universal design, is private, and is time effective. The proposed design solution utilizes alphabet clustering and implements name prediction as opposed to the linear search method discussed in the previous section. This solution proves to be more time effective for letter selection, and for 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 Nuance Communications’ XT9 (Nuance, 2009). 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 2 for a full example).

4.2 Cluster Selection

The alphabet is broken down into four clusters of five letters, and one cluster of six letters (Table 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 three 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 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 1. Standard Letter Clusters

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. 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.

4.3 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 2). 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 2).

4.4 Name Database

This prediction system for writing in a candidate's name is made possible through the use of a local database of names. A local database is utilized due to the ban of wireless devices and Internet connections in voting and tabulating machines according to the Voter Confidence and Increased Accessibility Act of 2009 (Holt, 2009 and VCIAA, 2009).

This database contains the most common names in the United States (Butler, 2005). 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.

4.5 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 can suggest "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. Furthermore, 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 2.

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 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**	CAR

	is L, P, or S	
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**

Table 2. Example Dialogue for Spelling Last Name, "Carlisle"

5 Experiment and Evaluation

The primary objective of this study was to observe and analyze how people interact with the predictive write-in system through speech. The goal of the study is to determine the time it takes a voter to use the write-in system developed. 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 their intended names.

5.1 Experimental Method

The participants were directed to fill out a pre-questionnaire to obtain their demographic information and prior usage with computing. Once the pre-questionnaire was completed, a scenario was given, introducing them to 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 to keep the results anonymous. 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. Data collected during the experiment included the name each participant chose to write-in and the times taken to spell that name.

5.2 Evaluation

A total of 40 participants participated in this study, of which more than 80 percent were undergraduates, Caucasians, and males. Presented in this section are calculated best-case comparisons between the predictive write-in method versus the linear search approach, as well as the experimental results from the study.

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. 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. The average time, for the names given, per letter was 1.09 minutes, with a standard deviation of 45 seconds.

Figure 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.

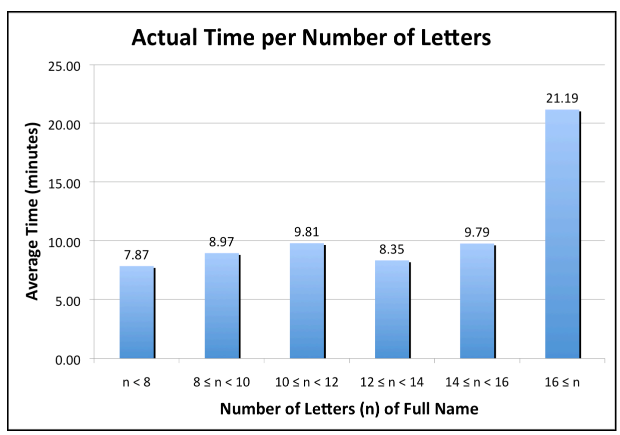

Figure 2. Average Time to Spell Full Names

Comparison: We calculated, at best case, how long it should take someone to spell the names from the study for both systems. 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. 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.

Figure 3 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 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. 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.

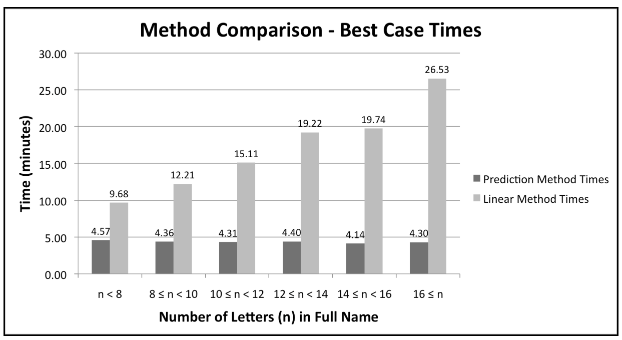

Figure 3. Best-Case Method Comparison of Times to Spell Full Names

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

Table 3. Calculated Predictive and Linear Method Statistics

6 Conclusion and Future Work

The ultimate goal of electronic voting systems today should be to allow anyone to vote privately and independently using a single design. The EAC 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 most disabilities, 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, using a predictive approach for efficiency.

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 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.

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. It may also be beneficial for future studies to include participants of a more diverse demographic, and 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.

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.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Grant Numbers CNS- 0540492 and 0837675. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

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

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

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

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

History of the Paper Ballot (2009) [online]. http://www.fairvote.org/righttovote/pballot.pdf.

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

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

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

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

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.

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

Prime III: One Machine, One Vote for Everyone. (2009) http://primevotingsystem.org/.

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

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:.

Write-in Candidate Requirements. Maine 2010 June Primary and November General Election Candidates (2010). [online] http://www.maine.gov/sos/cec/elec/2010/writeincandidateguide.doc.