ABSTRACT

Computational Modeling Reveals How Navigation Strategy and Ballot Layout Lead to Voter Error

by

Xianni Wang

Bad ballot design has affected the outcome of multiple elections in the United States. In order to build an automated tool for evaluation of ballots for potential usability problems, a range of voting behaviors on different ballot layouts have to be understood and modeled. The current studies are focussed on full-face paper ballots. Study 1 is an eye-tracking study. The ways that voters seek information on a full-face paper ballot was examined and the insights from the analysis results were integrated into Study 2. Study 2 is a cognitive modeling study. A family of 160 voting strategies were modeled using ACT-R to investigate how errors arise from the interaction of strategy and ballot design. The model was then validated by testing on a well-known bad ballot: the ballot from Kewaunee County, Wisconsin 2002. The Wisconsin error was reproduced successfully.

Acknowledgements

I am truly grateful to my advisor and mentor, Dr. Michael D. Byrne, for his patience as well as his endless support and guidance. I deeply appreciate the opportunity of being a part of the CHIL lab. I have learned so much from working with him, and simply cannot thank him enough.

I would like to thank Dr. Philip T. Kortum and Dr. David M. Lane for offering to be on my committee and for giving me invaluable feedback. I would also like to express my appreciation and gratitude to Dr. John K. Lindstedt for all of his kind help and his constant encouragement.

I would also like to sincerely acknowledge undergraduate researchers: Joshua Engels, Whitney Li, and Yoona Oh. I would not have been able to finish this research on time without their assistance.

Finally, a special thanks to my family, friends, and colleagues. Thank you so much for always being by my side and for believing in me all these years.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the United States, voting is an indispensable cornerstone of democracy and it provides citizens the right to express their preferences and make their voices heard. Anything that can cause the final tally to mismatch the intent of the voters is a threat to the integrity of the election. People usually think that filling out a ballot is a fairly simple task and it is unlikely to make mistakes throughout the process, nevertheless, voting errors are very common in almost all elections. There is strong evidence that the failure of the voting system—in particular, the design of the ballot—to accurately capture the voters' intent have affected the outcome of multiple elections in the United States (Laskowski et al., 2004).

Failure to capture voters' intent is essentially a usability problem. However, the usability of voting systems has received surprisingly little research attention over the years, and virtually none until the rear 2000. The majority of concerns on voting systems have been focused on election security. For example, some security researchers have been working on protection of the ballot or ballot box (e.g., preventing altering ballots or faking ballots) or the link from ballots to canvass (e.g., malicious alterations of the tally procedure). More recently, however, some consideration of the front end of this process has been undertaken by election security researchers, such as discussion of voter-verifiable paper audit trails (VVPATs). VVPAT is a method that allows election officials to confirm if the results collected by electronic voting systems accurately reflect the voters' intent. Meanwhile, it provides voters a second chance to verify if their choices were recorded correctly.

VVPATs are a step in the right direction in that the voter's role in the system has been considered, but even if the additional security procedures such schemes that require of voters are usable, the voting system still critically depends on correctly capturing the voter's intent. Failure of ballots to accurately capture the voters' intent threatens the integrity of the election just as much as a failure anywhere else in the chain. Even if the ballots and tally procedures are 100% secure, if the ballots themselves misrepresent the voters' intent, the election outcome will not necessarily represent the will of the voters. Therefore, it is very important for election officials to evaluate their ballot designs prior to deployment in election to avoid potential usability problems.

Ideally, conducting pre-election usability tests of every ballot would prevent error-producing ballot layouts. However, usability testing is normally time-consuming, expensive, and requires expertise, but many of the ballots have to be modified close to election day due to late changes (e.g., candidates who withdraw or are deemed ineligible). It is thus nearly impossible to evaluate thousands of ballot styles that deployed in every election all across the country through usability testing.

An automated evaluation system that does not require in-lab usability experiments and much time would be a great substitute for pre-election usability tests. The system should have the capability of assessing any ballot layout with a family of voting models that can simulate the entire space of possible voting behaviors. After running every ballot through each voting model repeatedly, the system will produce an assessment of whether the ballot is likely to lead to errors and at which parts of the ballot errors may occur. As a result, election officials would get the opportunity to modify their ballot designs accordingly prior to elections.

The goal of this thesis research is to expand the science necessary to support the development of the automated system for usability evaluation of ballots. This research builds upon several studies in which voting behaviors on different versions of single-race-per-screen ballots were learned and modeled. The two studies reported here are aimed to understand voting behaviors on full-face paper ballots, to expand the strategy space, and to develop human behavior models that cover a range of voting behaviors on full-face ballots.

1.1 Examples of Poor Ballot Design

Poor ballot design is a major usability problem in voting: some ballots fail to take into account human perceptual and cognitive limitations, and thus votes for the wrong candidate, as well as unintentional undervotes and overvotes, are very common in almost all elections. An undervote occurs when the number of votes is less than the maximum number allowed in a race. Voters have the right to undervote if they choose to do so. An overvote occurs when the number of votes is more than the maximum number allowed. The race that is overvoted cannot be counted in the final tally.

One of the most infamous error-inducing ballots is the "butterfly ballot" from Palm Beach County, Florida during the 2000 election (Wand et al., 2001; see Figure 1.1). This ballot presented candidates in the presidential race across two columns; the democratic candidates are listed second on the left, but they correspond to the third hole on the ballot. As a result of the inconsistency, more than 2,000 votes intended for Gore were cast for Buchanan instead.

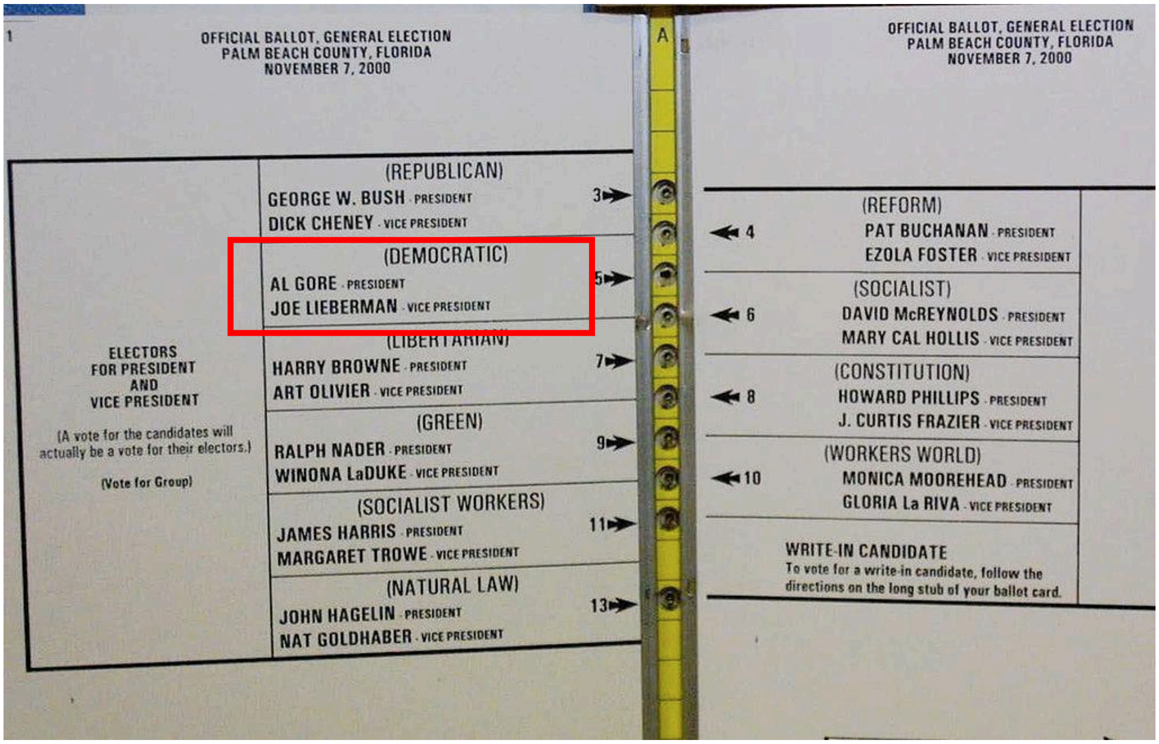
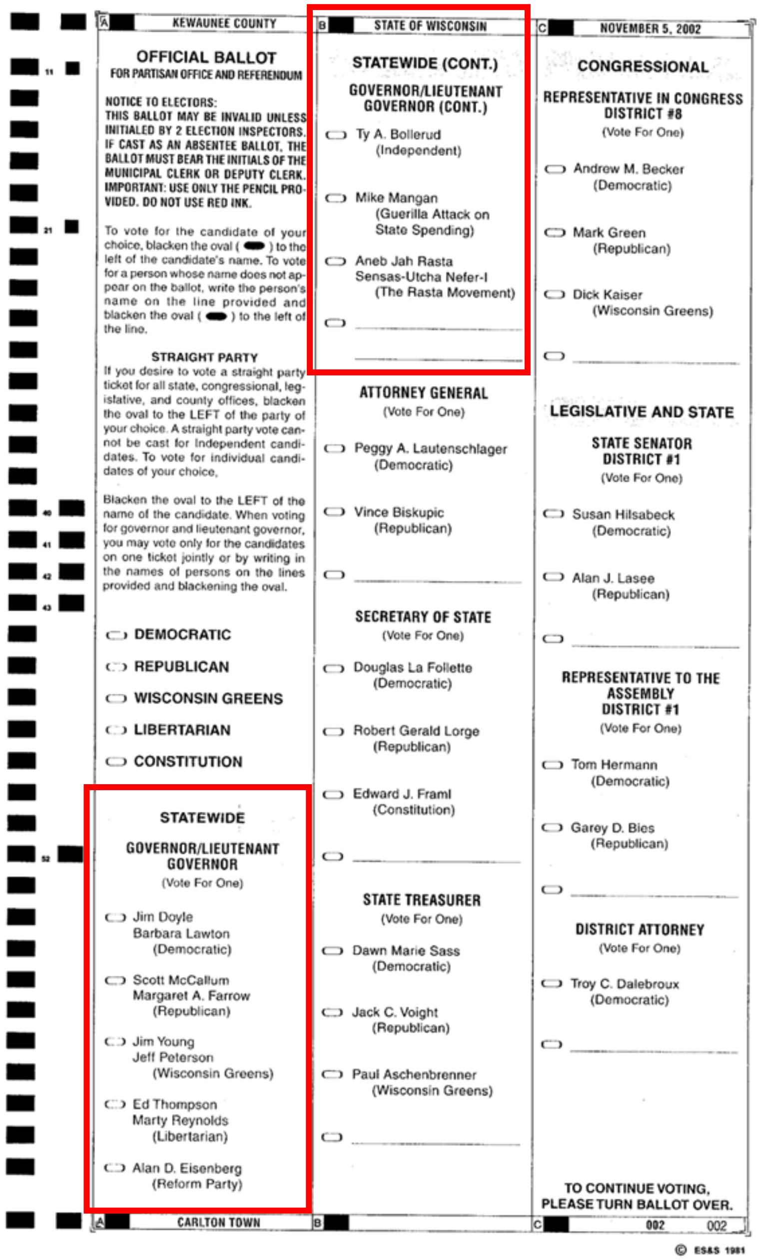


Figure 1.1 : The "butterfly ballot" from Palm Beach County, Florida 2000. The democratic candidates are listed second on the left, but they correspond to the third hole in the middle. Many voters who intended to vote for democratic candidates ended up filling out the second hole.

Another example of poorly designed ballot is the ballot used in Wisconsin in 2002, which led to many unintentional overvotes (see Figure 1.2). On this ballot, the race for Governor was split across two columns. Many voters considered the two sections as representing two races, causing them to vote twice, once in each column, thus rendering an invalid vote rate of 11.8% for this race, in contrast to an invalid vote rate of 1% for this race statewide (Norden et al., 2008).


KEWAUNEE COUNTY | STATE OF WISCONSIN | NOVEMBER 5, 2002

OFFICIAL BALLOT
FOR PARTISAN OFFICE AND REFERENDUM

NOTICE TO ELECTORS:
THIS BALLOT MAY BE INVALID UNLESS INITIALED BY 2 ELECTION INSPECTORS. IF CAST AS AN ABSENTEE BALLOT, THE BALLOT MUST BEAR THE INITIALS OF THE MUNICIPAL CLERK OR DEPUTY CLERK. IMPORTANT: USE ONLY THE PENCIL PROVIDED. DO NOT USE RED INK.

To vote for the candidate of your choice, blacken the oval ( ) to the left of the candidate's name. To vote for a person whose name does not appear on the ballot, write the person's name on the line provided and blacken the oval ( ) to the left of the line.

STRAIGHT PARTY
If you desire to vote a straight party ticket for all state, congressional, legislative, and county offices, blacken the oval to the LEFT of the party of your choice. A straight party vote cannot be cast for Independent candidates. To vote for individual candidates of your choice,

Blacken the oval to the LEFT of the name of the candidate. When voting for governor and lieutenant governor, you may vote only for the candidates on one ticket jointly or by writing in the names of persons on the lines provided and blackening the oval.

DEMOCRATIC
REPUBLICAN
WISCONSIN GREENS
LIBERTARIAN
CONSTITUTION

STATEWIDE
GOVERNOR/LIEUTENANT GOVERNOR
(Vote For One)
Jim Doyle
Barbara Lawton
(Democratic)
Scott McCallum
Margaret A. Farrow
(Republican)
Jim Young
Jeff Peterson
(Wisconsin Greens)
Ed Thompson
Marty Reynolds
(Libertarian)
Alan D. Eisenberg
(Reform Party)

STATEWIDE (CONT.)
GOVERNOR/LIEUTENANT GOVERNOR (CONT.)
Ty A. Bollerud
(Independent)
Mike Mangan
(Guerilla Attack on State Spending)
Aneb Jah Rasta
Sensas-Utcha Nefer-I
(The Rasta Movement)

ATTORNEY GENERAL
(Vote For One)
Peggy A. Lautenschlager
(Democratic)
Vince Biskupic
(Republican)

SECRETARY OF STATE
(Vote For One)
Douglas La Follette
(Democratic)
Robert Gerald Lorge
(Republican)
Edward J. Framl
(Constitution)

STATE TREASURER
(Vote For One)
Dawn Marie Sass
(Democratic)
Jack C. Voight
(Republican)
Paul Aschenbrenner
(Wisconsin Greens)

CONGRESSIONAL
REPRESENTATIVE IN CONGRESS DISTRICT #8
(Vote For One)
Andrew M. Becker
(Democratic)
Mark Green
(Republican)
Dick Kaiser
(Wisconsin Greens)

LEGISLATIVE AND STATE
STATE SENATOR DISTRICT #1
(Vote For One)
Susan Hilsabeck
(Democratic)
Alan J. Lasee
(Republican)

REPRESENTATIVE TO THE ASSEMBLY DISTRICT #1
(Vote For One)
Tom Hermann
(Democratic)
Garey D. Bies
(Republican)

DISTRICT ATTORNEY
(Vote For One)
Troy C. Dalebroux
(Democratic)

TO CONTINUE VOTING, PLEASE TURN BALLOT OVER.

CARLTON TOWN | 002 002

© ES&S 1981

Figure 1.2 : The ballot from Kewaunee County, Wisconsin 2002. The gubernatorial race was split across two columns. Many voters considered them as two different races and voted twice.

A more recent example is the ballot used in Broward County, Florida in 2018, which caused many unintentional undervotes. As can be seen in Figure 1.3, there were two races placed beneath the ballot instructions. Voters who skipped over the instructions were in danger of skipping past these two races to the top of the middle column. Therefore, this ballot design caused more than 26,000 undervotes in the Senate race, where the margin of victory was about 13,000 votes (Appel, 2018).

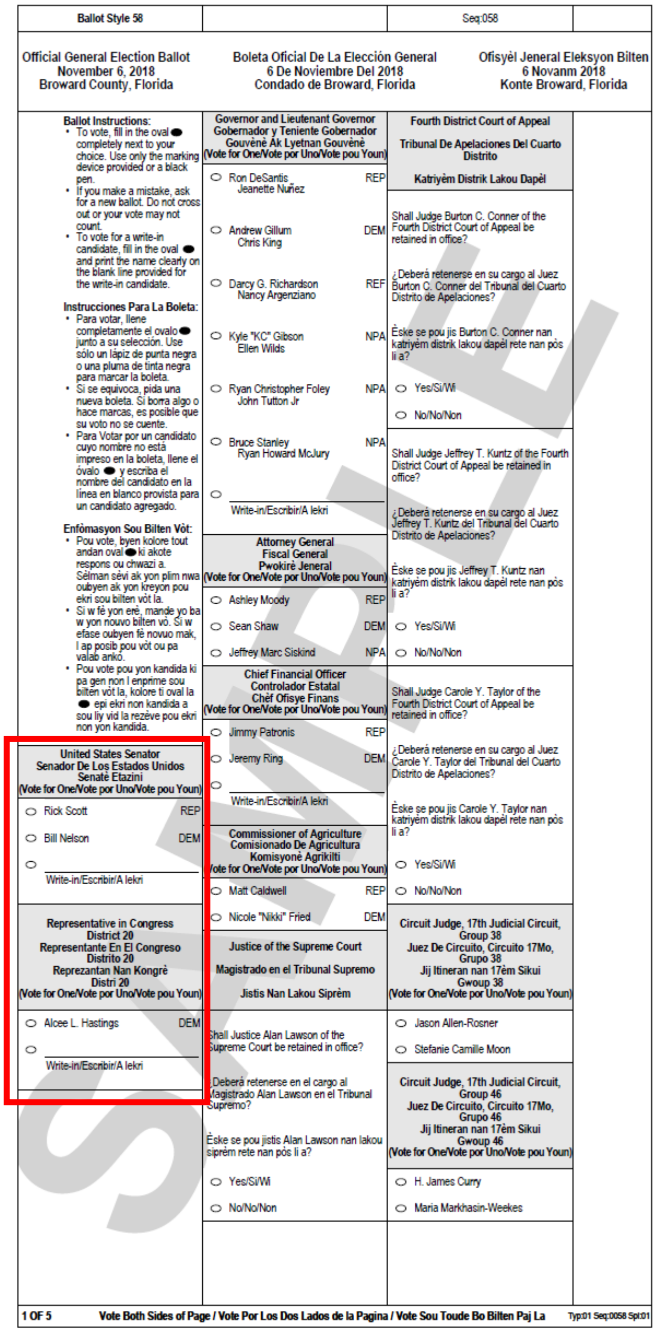


Figure 1.3 : The ballot from Broward County, Florida 2018. There were two races placed beneath the ballot instructions. Some voters who skipped over the instructions skipped past these two races to the top of the middle column.

1.2 Predicting Errors in Filling out Ballots

A standard usability evaluation prior to deployment would likely detect most poor ballot designs and prevent most errors. However, few usability specialists have been asked to perform such tests prior to an election. Instead, election officials, who have little formal training or expertise in assessing usability, are left to the task. In addition, most elections in the United States are administered at the county level, and there are over 3,000 counties in the country. Within each county, there are often hundreds of different precincts, each with a slightly different ballot style. Thus, for each national election, tens of thousands of ballot designs are deployed. The scale of this problem makes conducting a traditional usability test for every single ballot intractable.

Because it is impossible to perform traditional usability testing on every ballot before every election deployment, examining ballots through an automated error prediction tool would be a preferable solution. Error prediction methods are often based on traditional hierarchical task models (e.g., Annett & Duncan, 1967), which often assume that the processing system is explicitly hierarchical in structure and therefore break down complex tasks into hierarchies and sub-goals. Botvinick and Plaut (2004) suggested that hierarchical schemas and goals are not always necessary, at least in routine behavior. Instead, they presented a recurrent network model that uses recurrent connections within a network, which map from environmental inputs to action outputs, to represent an everyday task. However, Cooper and Shallice (2006) contrasted this recurrent network model with their more traditional, hierarchically structured interactive activation model. They criticized Botvinick and Plaut’s recurrent network approach, describing a set of problems with the approach, such as its behavioral inflexibility, and concluded that hierarchical structures are still necessary and play a causal role in the control of behavior.

Another approach to studying human error is to create human performance models using ACT-R (Anderson, 2007). This goes one step beyond models based on a traditional hierarchical structure by using cognitive architectures. ACT-R is a cognitive architecture that simulates and integrates human cognition, attention, and motor behavior. Generally speaking, an ACT-R model of a task consists of both the architecture and the requisite knowledge to perform the specific task, and it is often connected to a simulation of the environment in which the task is performed or the actual software that humans use to perform the task. Therefore, ACT-R can help researchers to understand how people organize knowledge and produce behavior in different ways.

However, it is not easy to predict voting errors using ACT-R. First, in general, ACT-R models are fitted to and/or make predictions about average human behavior. However, predicting voting errors cannot simply be a question of fitting the mean, because even if the average person does not make an error, there may still be a substantial number who do. Second, there are many types of voting errors: unintentional overvotes and undervotes, filling in a wrong bubble, etc. Therefore, it takes time and effort to create models that cover all possible errors.

1.3 Using ACT-R to Model Voting Behaviors

Using ACT-R to study human performance and errors is not a new idea. There are several published ACT-R models that can make the same errors as people (e.g., Anderson et al., 1998; Halbrügge et al., 2015; Lebière et al., 1994; Trafton et al., 2011). Some error prediction models have also been developed for voting tasks.

1.3.1 Greene's Model

In Greene (2010), an ACT-R model was presented that could sometimes make the same mistake that some voters ever made. In 2006 election, voters in Sarasota, Florida voted through direct recording electronic voting systems (DRE) that do not need physical ballots and can record votes directly onto computer memory devices. The incident occurred in the Congressional election. As can be seen in Figure 1.4, there was a single race present on the first DRE screen, but there were two races listed on the second screen. Also, the race heading "Congressional" was not present above the US Representative race as it was on the first screen. This layout inconsistency led to an undervote of 13.9% (about 18,000 votes) for the US Representative race, where the margin of victory was about 380 votes.

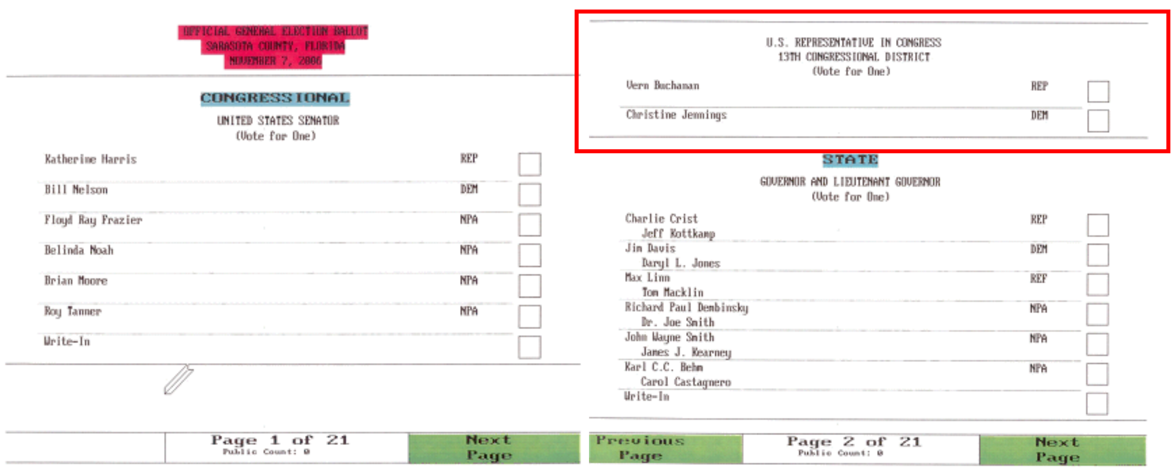


Figure 1.4 : Two screen captures from the 2006 Sarasota County electronic voting system. First screen is on the left (one race presented), second screen is on the right (two races presented). Many voters failed to vote for the US Representative race that displayed on the top of the second screen.

Greene (2010) reproduced these two screens and modeled two voting strategies. The first strategy was to read the first screen from top to bottom before selecting a candidate, and then to recall a useful location from the first screen to use to direct the visual search on the next screen. With this strategy, the model used the first screen to set expectations about where to find relevant landmarks (e.g., titles of races); it could then miss the critical top race on the second screen when the model extended those expectations from one screen to the next. The second strategy was to read both screens from top to bottom, without any recall. In contrast to the first strategy, the second strategy did not result in a critical top race undervote.

At the first glance, it seems that the header highlighting was the cause of the undervotes. However, Greene (2010) came to the opposite conclusion—undervote error rates were actually greater with header highlighting than without. In addition, Greene's studies suggested that the presentation of multiple races had a significant effect on the undervotes in the US Representative race; the interaction of poor ballot design and voting strategies played an important role in causing the undervotes.

1.3.2 Multi-Strategy Model

Greene's (2010) model offers a meaningful opportunity for computational human performance modeling to make a unique contribution to the voting field. However, this model does not reflect the full complexity of voting. Different voters almost certainly approach ballots differently. It is therefore critical that the models reflect not just one or two voting strategies, but the entire range of voting behaviors, so that specific interactions between voting strategies and ballot designs can be uncovered. To capture more of the voting complexity, a family of 40 ACT-R models of a voting task was developed (Wang et al., 2019). For each model, the memory strategy, ballot knowledge, and visual search strategy were considered independently.

Memory strategy represents how voters access their memories when they cast a vote. Voters have to remember their choices, and they access their memories in different ways. There are two primary memory strategies for simple form-filling tasks like voting: recall and recognition. Some voters can simply recall the names of those for whom they intend to vote, at least in some races. For example, many voters, when prompted, can recall from memory the candidate for whom they intend to vote in presidential elections. Other voters may instead scan the list of names first to try to recognize their preferred candidates. Some voters vote almost exclusively according to party affiliation but then have to remember which races, if any, have exceptions. Some voters may rely on party affiliation if they can neither recall for whom they intended to vote nor recognize any of the candidates' names on the list. Some voters may also write out a list and bring it into the voting booth, although it is not clear how common this is, and it is, in fact, illegal in some jurisdictions. Note that in this scenario, the voters effectively use a recall strategy as well. They just have a 100% success rate of recall for every race.

Ballot knowledge refers to voters' level of knowledge of the races and candidates. Voters have different levels of knowledge about the races and candidates: some voters might have encoded all of the candidates' names, some may only know the names of candidates they intend to vote for, and some may only have partial lists of the intended candidates' names in their memories. In addition, ballot knowledge is not always easy to recall. Some voters may only remember their choices for the first few races because it is much more likely that voters will have more frequent exposure to top-of-the-ballot candidates. ACT-R represents situations like this using base-level activation, which reflects the recency and frequency of a specific memory.

Visual search strategy indicates voters' visual directions when conducting a visual search. While reading in a serial order is the most common search strategy, eye-tracking studies have demonstrated that it is not universal (Aaltonen et al., 1998; Fleetwood & Byrne, 2006). People scan displays in different ways: some readers read in a serial, item-by-item pattern, from one corner to its diagonal opposite; some people scan globally and read all the bold, large, or colored headers first; and some simply prefer to scan randomly. Furthermore, humans have a remarkable ability to organize their perceptual inputs. The human visual system tends to group individual items in a visual image into larger structures under certain circumstances. This allows for the more efficient use of attention but sometimes leads to critical errors in executing a task. For example, the poor ballot used in Wisconsin in 2002 misled the human visual system, which caused unintentional overvotes in the race for Governor. To handle situations like this, the multi-strategy model makes use of a visual grouping algorithm that enables more realistic visual scanning behaviors (Lindstedt & Byrne, 2018).

1.3.2.1 The Voting Task

To simulate the various abovementioned human performance in an emulated voting task, the models voted on a version of the VoteBox task environment. VoteBox is a DRE platform developed by Sandler, et al. (2008); it is a research platform that helps to investigate both usability and security issues in voting systems, and multiple experiments have been published in which human subjects voted using VoteBox (e.g., Everett, 2007; Everett et al., 2008). The voting task consists of 21 races that share a consistent layout (see Figure 1.5). The layout was designed to be easy to understand, with a relatively simple display that comprised the voting instructions, title of the race, candidates' names and party affiliations, and "previous page" and "next page" buttons, all clearly arranged and presented.

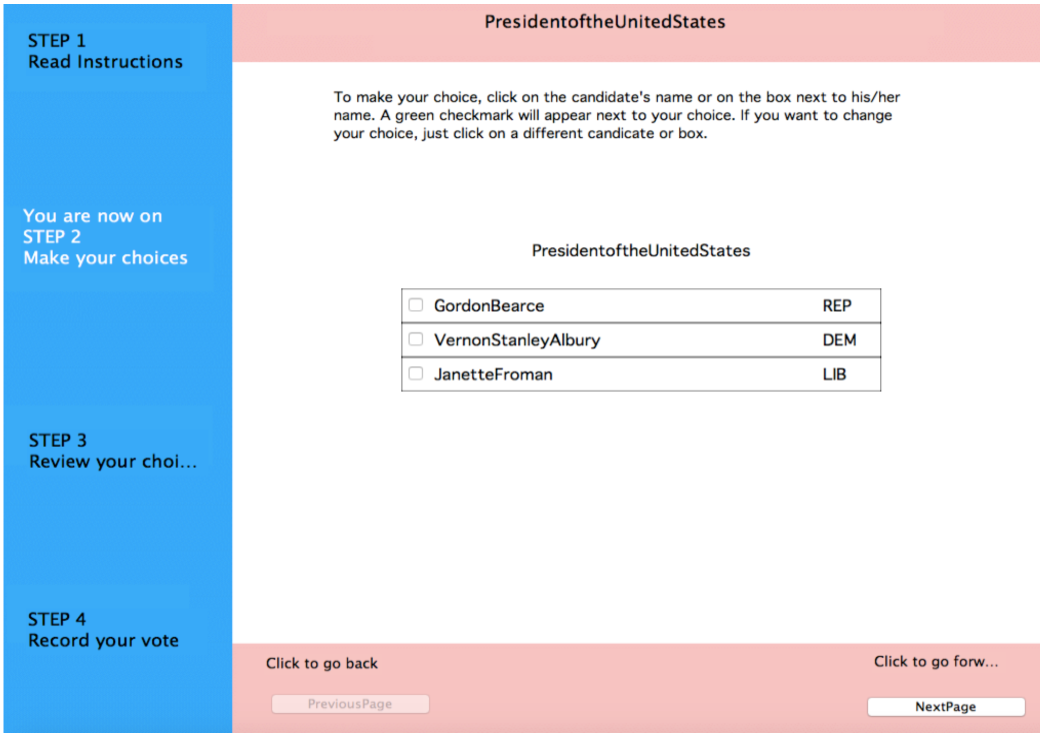


Figure 1.5 : VoteBox mock ballot of a Presidential race. All elements (voting instructions, title of the race, names, and party affiliations) were clearly arranged and displayed.

All versions of the voting model go through two phases. The first is a studying phase in which the model studies the display thoroughly to retain group information produced by the visual grouping algorithm (see Figure 1.6). First, the grouping algorithm takes an unexamined point from the screen as the starting point and assigns it to the current group. Then, the algorithm examines and adds any other points within a pre-determined grouping radius to the current group. Next, it repeats the previous step for each new point added and keeps growing the current group, until no unexamined points remain within the radius of any group members. After that, it selects another unexamined point as a new starting point, and just repeats the entire process until all points in the scene have been assigned to a group.

The second phase is a voting phase; after obtaining and storing group information during the first phase, the model now has expectations about where to look. It directs its “gaze” to the appropriate place and then makes a vote. In this phase, the models may perform differently because each model simulate a specific combination of memory strategy, ballot knowledge, and visual search strategy.

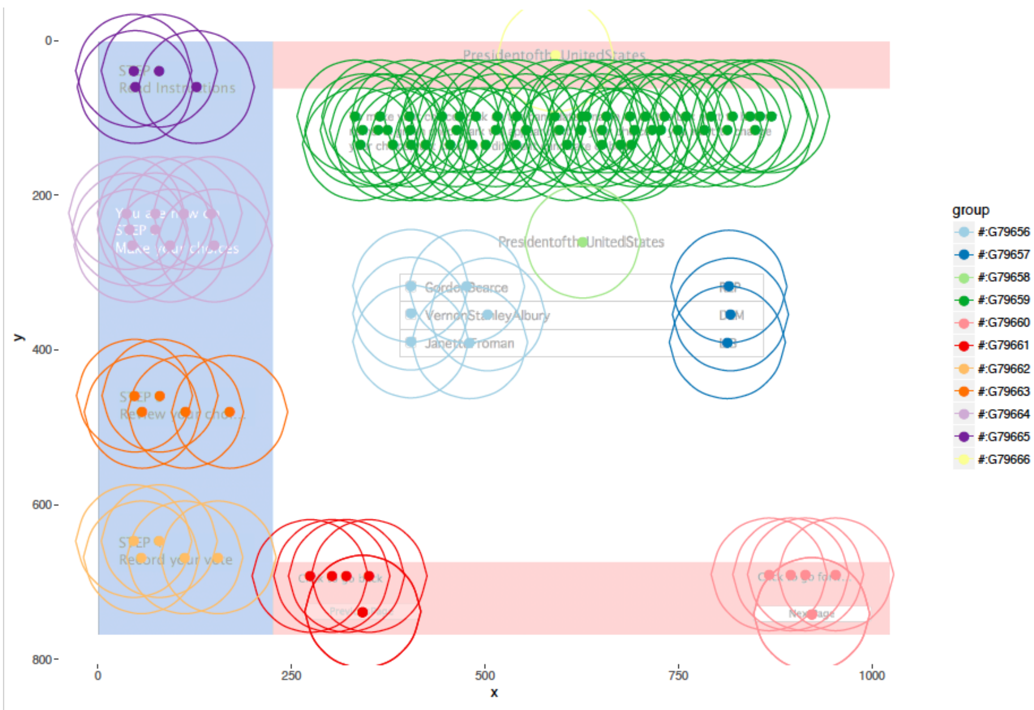


Figure 1.6 : The outcome of the visual grouping algorithm. 11 visual groups (indicated by different colors) are identified in this mock ballot layout.

1.3.2.2 Modeling Strategies

To produce a comprehensive error prediction, four memory strategies, five levels of ballot knowledge, and two visual search strategies were defined. First, the models capture four memory strategies that one could reasonably expect a voter to employ: a strictly retrieval-based strategy, a strictly recognition-based strategy, a retrieval-then-recognition contingency strategy, and a simple party-only look-up strategy (in case of exceptions to the voter's default party). The first strategy represents the scenario in which the model first tries to retrieve the candidate's name from memory. If the model fails to recall the name, then it relies simply on party affiliation. The second strategy considers the situations in which the model first tries to retrieve the candidates' name, but, if the retrieval fails, it then scans the list of names and votes for the one it recognizes. If recognition also fails, it votes by party affiliation. For the third strategy, the model does not even attempt to retrieve; rather, it scans the list of names to see if it can recognize any of them. If recognition fails, it votes by party. For the last strategy, the model simply votes based on party affiliation. It first retrieves the specific party affiliation for particular races, but, if the retrieval fails, default party affiliation becomes the criterion. The last step of these four memory strategies—voting by default party affiliation—is used only when all the previous steps fail.

Second, five levels of ballot knowledge were created (see Table 2.1). First, there are three levels of how many candidates' names were stored. The models could remember all candidates' names, only the intended candidates' names, or only the first 70% of the intended candidates' names. Then, two types of activations for intended candidates were assigned: roll-off activations and constant high-level activations. Models with roll-off activations are most familiar with the candidates for the first several races; then, as they progress down the ballot, their familiarity with candidates decreases. In the second condition—constant high-level activations—the models are highly familiar with all races to the same degree. Note that the various contents and activation levels of memory were not chosen as an exhaustive search of all possible knowledge held by voters, but rather as an illustrative sample of common voter scenarios—some voters have certainly done their homework extensively, while others have likely only decided "important" races.

Finally, two visual search strategies were used when looking for candidates: a serial search and a random search. The serial search strategy is a serial item-by-item search with a left-to-right, top-to-bottom pattern. With the random search strategy, the models conduct a random search.

Ballot knowledge	Candidates' names	Activations for intended candidates
FULL-MEMORY	All candidates	Races 1 to 7: 0.7 Races 8 to 14: 0.6 Races 15 to 21: 0.5
ALL-ROLLOFF	Intended candidates only	Races 1 to 7: 0.7 Races 8 to 14: 0.6 Races 15 to 21: 0.5
ALL-PERFECT	Intended candidates only	All races: 0.8
MOST-ROLLOFF	70% of intended candidates	Races 1 to 3: 0.8 Races 4 to 7: 0.7 Races 8 to 11: 0.6 Races 12 to 15: 0.5 Race 16 to 21: Abstained
MOST-PERFECT	70% of intended candidates	Race 1 to 15: 0.8 Race 16 to 21: Abstained

Table 1.1 : Five levels of ballot knowledge.

1.3.2.3 Model Evaluation

After developing 40 voting models that crossed four memory strategies with five levels of ballot knowledge and two visual search strategies, each voting model was tested multiple times for model evaluation or, in other words, error prediction. Then, the average across those runs was calculated. A 5% overall error rate generated by the model was expected, and, for the model predictions, the 95% confidence intervals were desired to be no wider than 2% in either direction. The table in Byrne (2013) shows that 457 model runs are required; to be slightly more conservative, 500 runs per model were performed.

For each model run, the ballot, as completed by the model, was compared with the "intent" initialized at the beginning of the run, and any discrepancies were noted as errors. Errors occurred across the entire voting process. The model might have retrieved an unintended name, recognized an unintended name, or failed to retrieve and then recognized an unintended name. For the model that simply made votes based on party affiliation, it may have retrieved an unintended party. The model may even have failed to retrieve and/or recognize an intended name, and then have voted by default party affiliation. Democratic was used as the default party affiliation for this model evaluation; however, intended candidates' party affiliations did not always match the default party affiliation. The model occasionally also mis-clicked on the name above or below the intended name.

Overall, the models generated an average 5% error rate across all voting models. Differences in error rates with visual strategies were not found, which means that using either a serial or a random scanning pattern did not affect the voting results. The main story in these results is about memory strategy and ballot knowledge. Differences in voting errors based on the interaction between voting strategy and ballot knowledge was observed.

Figure 1.7 presents five groups of bars that represent the error rates of the five levels of ballot knowledge. For each level of ballot knowledge, the percentages of the errors for the four memory strategies are displayed. For the FULL-MEMORY condition, the model generated 9% more errors than the other four levels of ballot knowledge. The model also generated more errors with roll-off activations for intended candidates. For the MOST-ROLLOFF and ALL-ROLLOFF conditions, the voting model was 2% more likely to make errors than with the MOST-PERFECT and ALL-PERFECT conditions. Additionally, for the four levels of ballot knowledge other than FULL-MEMORY, there were clearly fewer errors with the three-step “retrieve-recognize-party” memory strategy.

To determine which process the model was using when it made an error, the error attributions for each vote were also recorded and analyzed. In Figure 1.7, each bar is partitioned into three colors, which represent the three processes the model could have been using when it made an error: retrieval, recognition, or party affiliation. For the FULL-MEMORY condition, most of the errors occurred in the recognition and/or retrieval processes. Within FULL-MEMORY, 7% more voting errors were generated with the “recognize-party” memory strategy. However, for the other four levels of ballot knowledge, differences in error attributions with memory strategies were not apparent; most of the errors were generated in the last steps—voting by party affiliation.

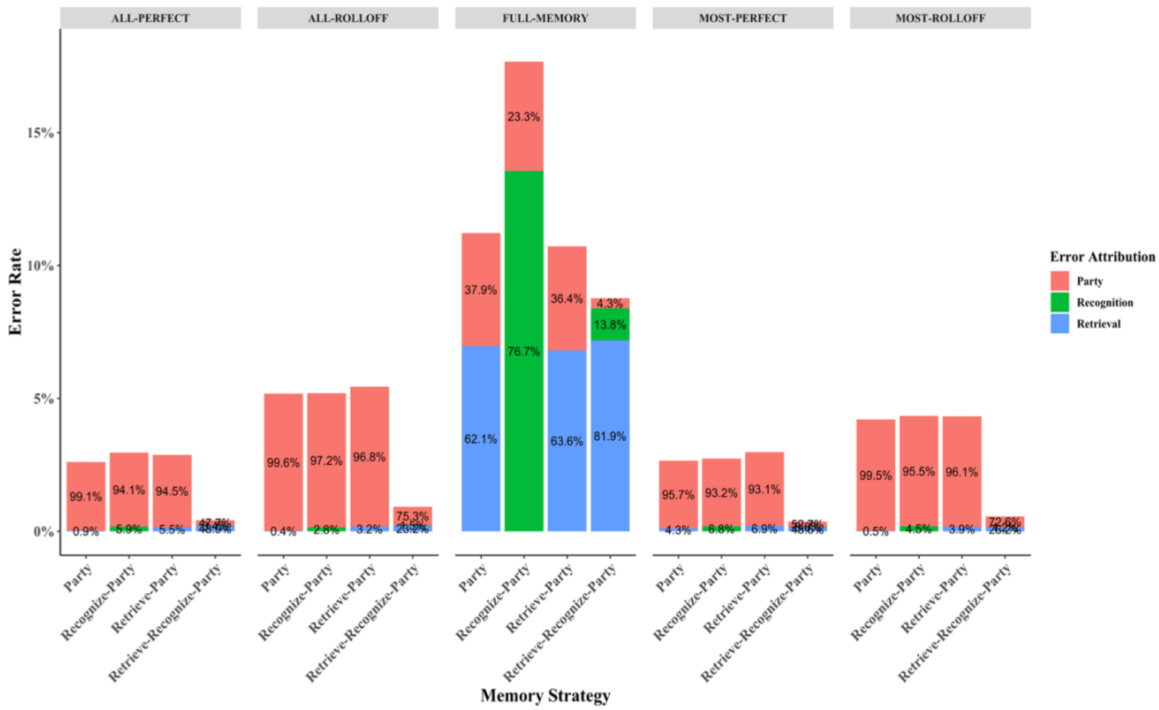


Figure 1.7 : Interaction between memory strategy and ballot knowledge in voting error rates. The bars show voting errors, grouped by ballot knowledge levels (labeled at the top); each group shows the error rates for the four memory strategies. The colors of the bars indicate the processes the model was using when it made an error. Red, green, and blue represent the party affiliation, recognition, and retrieval, respectively.

1.3.2.4 The Analysis of Error Predictions

The error predictions reveal that extra ballot knowledge actually led to more errors, especially with the involvement of recognition. Common sense would suggest that a broader knowledge base should help to mitigate mistakes, but this is not always the case. Consider the recognition heuristic (Goldstein & Gigerenzer, 2002). The recognition heuristic describes a situation where, if one of two objects is recognized and the other is not, the recognized object is more likely to be selected. This strategy requires ignorance to make a choice—if people know everything or nothing about the options, it simply does not work. For example, for the question “which city has a larger population?” most people choose Dublin over Nenagh since they can recognize Dublin only. However, it is harder for people to make a selection if the choices become San Diego and San Antonio, as they are more likely to recognize both of these cities. Similarly, in the voting task in this study, the models knew everything in the FULL-MEMORY condition, including both the intended and unintended candidates' names. Thus, compared to the other four levels of ballot knowledge, the memory strategies did not work well with FULL-MEMORY, and more errors occurred in the recognition processes.

Because of the more frequent recognition errors, a greater impact of candidate name order can be expected with the FULL-MEMORY condition. Voters who cannot recall their intended candidate's name must scan the list of names and see if they can recognize any, and their choices can be biased by the order in which candidates' names appear on the ballot (Miller & Krosnick, 1998). Similarly, the model with "recognize-party" memory strategy checks each candidate, sees if it recognizes the name, and if so, votes for it. Thus, since some voters do use top-to-bottom visual search, an advantage for the top candidate can be predicted.

Another finding has to do with the interaction between task knowledge and recall performance. Schooler and Anderson (1997) suggested an association between the number of choices and recall performance, positing that the more choices we have, the more likely we are to make a recall error at each name. The same relationship can be observed in the models: the FULL-MEMORY condition contains both intended names and unintended names, and the models could either retrieve an intended name or an unintended name for each race in that condition; it was therefore more likely to make errors in the retrieval process since incorrect answers are available. However, with the other four levels of ballot knowledge, there are only intended names available in memory. Wrong names were therefore less likely to be retrieved with these four levels of knowledge.

The error predictions also indicate that the three-step “retrieve-recognize-party” memory strategy had a better performance than the two-step memory strategies. As can be seen in Figure 1.7, a large portion of the errors came from the last steps, voting by party affiliation, across five levels of knowledge. Comparing to the two-step strategies, the additional one step prevented errors that could be made in the last step, so the least amount of errors was generated with the three-step memory strategy.

Note that the errors made in these models are not the result of poor ballot design. However, the interactions between strategy, knowledge, and ballot design should show how the visual layout of the ballot could influence error rates. Poor layouts may not induce all voters into error, but differentially affect those who use particular strategies. A minor difference in memory strategies, visual strategies, or ballot knowledge can often yield different results. For example, even if every voter applies an identical memory strategy and visual strategy, the voters who are familiar with all candidates’ names on the ballot are more likely to make recognition errors than the voters who are simply familiar with the intended candidates’ names.

The multi-strategy model represents the first use of ACT-R as an error prediction tool to diagnose whether there are particular combinations of strategies that lead to error. The results of the error prediction demonstrate that subtle interactions between strategy and knowledge can have substantial effects on error rates. It is therefore critical to consider multiple combinations of both when attempting to model errors, even in a task that appear as simple as voting.

1.4 Studying Voter Reading Patterns

In addition to the modeling work for voting tasks, an eye-tracking study was conducted to study voting behaviors (Lindstedt et al., 2019). While a great deal of cognitive activity can be inferred from observations of standard behavioral traces (using measures of reaction and response latencies, accuracy, sequencing, etc.), eye-tracking techniques can help researchers more precisely examine the flow of information from the task into the cognitive system. As discussed earlier, both recognition and recall strategies are reasonable approaches to the voting task, as each seems sufficiently adapted to the task to succeed. In addition, even small differences in error rates between the two strategies can be impactful because of the large scale and tight margins of many elections. As a result, to better assess the extent to which these differentially effective strategies are actually employed while filling out a ballot, eye-tracking data was collected.

The voting task in this study emulates the task designed for evaluation of the multi-strategy model mentioned above. As can be seen in Figure 1.8, it consists of 21 races, and the layout is very similar to the VoteBox task (see Figure 1.5). Sixteen participants were instructed to select candidates, either using a guide to the fictitious candidates' policies or a simple "slate" instructing them to vote for specific individuals.

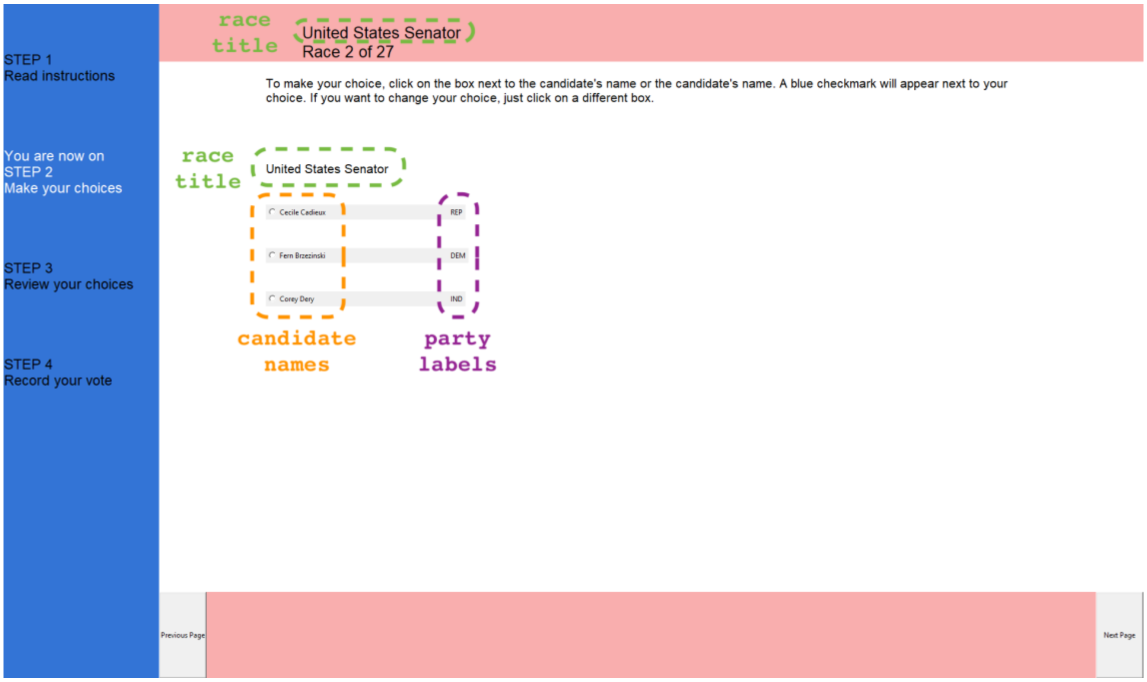


Figure 1.8 : The VoteBox emulator task. It emulates the task designed for the evaluation of the multi-strategy model (see Figure 1.5). Three regions of interest (ROIs) for the eye tracking analysis are indicated by the dotted lines.

The eye-tracking data was analyzed with a focus on three regions of interest (ROIs) that participants fixated on before selecting a candidate for each race: race title, candidates' names, and party affiliation. If a participant looked at the race title as well as the candidates' names or the party affiliation, a retrieval strategy was considered to have been used; if a participant did not look at the race title, but did look at the candidates' names or party affiliation, a recognition strategy was presumed to have been used; if a participant had fixations only on the race title, or only on non-ROI areas of the screen, the memory strategy was considered as unclear.

This study provides evidence that voters utilize both recall and recognition memory strategies when voting—overall, participants employed a retrieval-based strategy in 54.8% of trials, a recognition-based strategy in 33.3% of trials, and the strategy was unclear in 11.9% of trials. The results of this study also suggest that some participants switched between the two strategies while filling out a ballot, although each participant appeared to have preferred one strategy over another (see Figure 1.9).

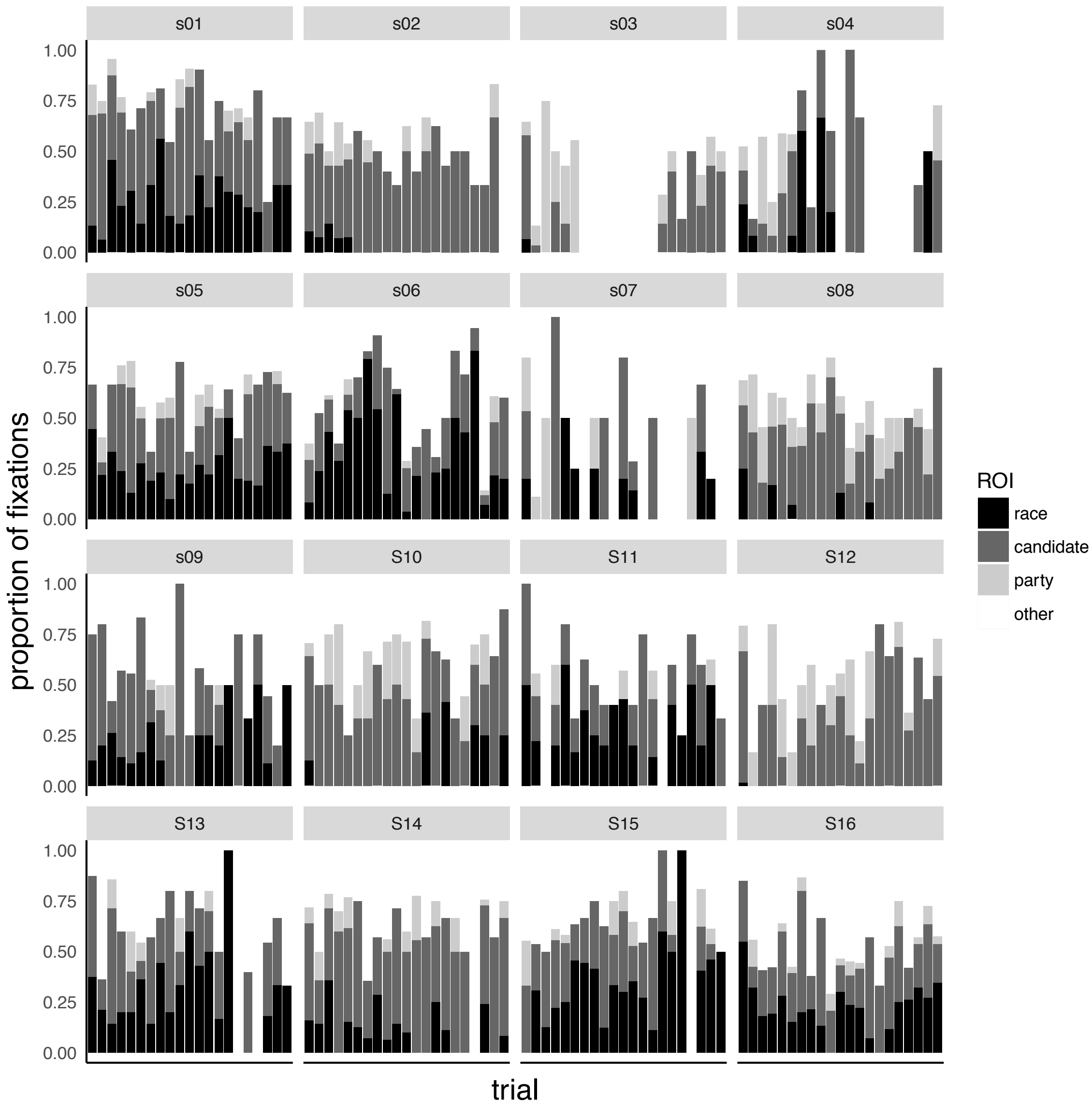


Figure 1.9 : The proportion of total fixations (y-axis) for each of the three main ROIs (shade) for each trial (x-axis, first trial on left) completed by each participant. Each stacked bar represents the breakdown of a participant's fixations of a single trial. Missing data or non-ROI fixations are in white.

It is also notable in Figure 1.9 that some participants fixated only on the party affiliations for several trials; some of the time participants were employing a strictly party-based memory strategy. Furthermore, this study provides a good support for the multi-strategy model; it proves that the idea that one can understand the error space by investigating only one strategy or predicting mean behavior is likely to miss critical combinations of factors that produce errors.

1.5 Research Design

The aforementioned eye-tracking study demonstrates the importance of covering the entire strategy space when developing cognitive models, but the voting task was conducted using a single-race-per-screen ballot. Paper ballots, with many races on a single display, are fundamentally different: voters must navigate both between and within races on one ballot. To eventually create an error prediction tool to detect the potential usability problems of paper ballots, it is necessary to understand how people vote using them.

The current research consists of two studies. Study 1 is an eye-tracking study in which data were collected for a full-face ballot voting task. The collected eye-tracking data, such as the reading patterns of the participants, were analyzed to identify the ways in which voters seek information on a full-face ballot, and insights from the analysis results were integrated into the second study.

Study 2 is a cognitive modeling study that expanded the strategy space covered by the multi-strategy model. First of all, a family of voting models that cover different navigation strategies was developed. Next, for model evaluation, the models were tested on a group of randomly generated full-face ballots, and the ways in which errors emerged from the interaction of strategy and ballot design were thus identified. Finally, for model validation, the models were tested on a well-known poor ballot: the Wisconsin ballot (see Figure 1.2).

Chapter 2

Study 1

The aim of Study 1 is to understand how voters interact with a full-face paper ballot. This study went a step beyond the eye-tracking study previously described in section 1.4: it sought not only to examine what information from the display a participant includes in his/her approach to completing the task, but also to identify how participants navigate through the races, party affiliations, and candidates together among the multiple races on one screen. By using an eye-tracking system and a custom-built ballot interface, the interactions of the participants with the ballot and the patterns among those interactions were recorded.

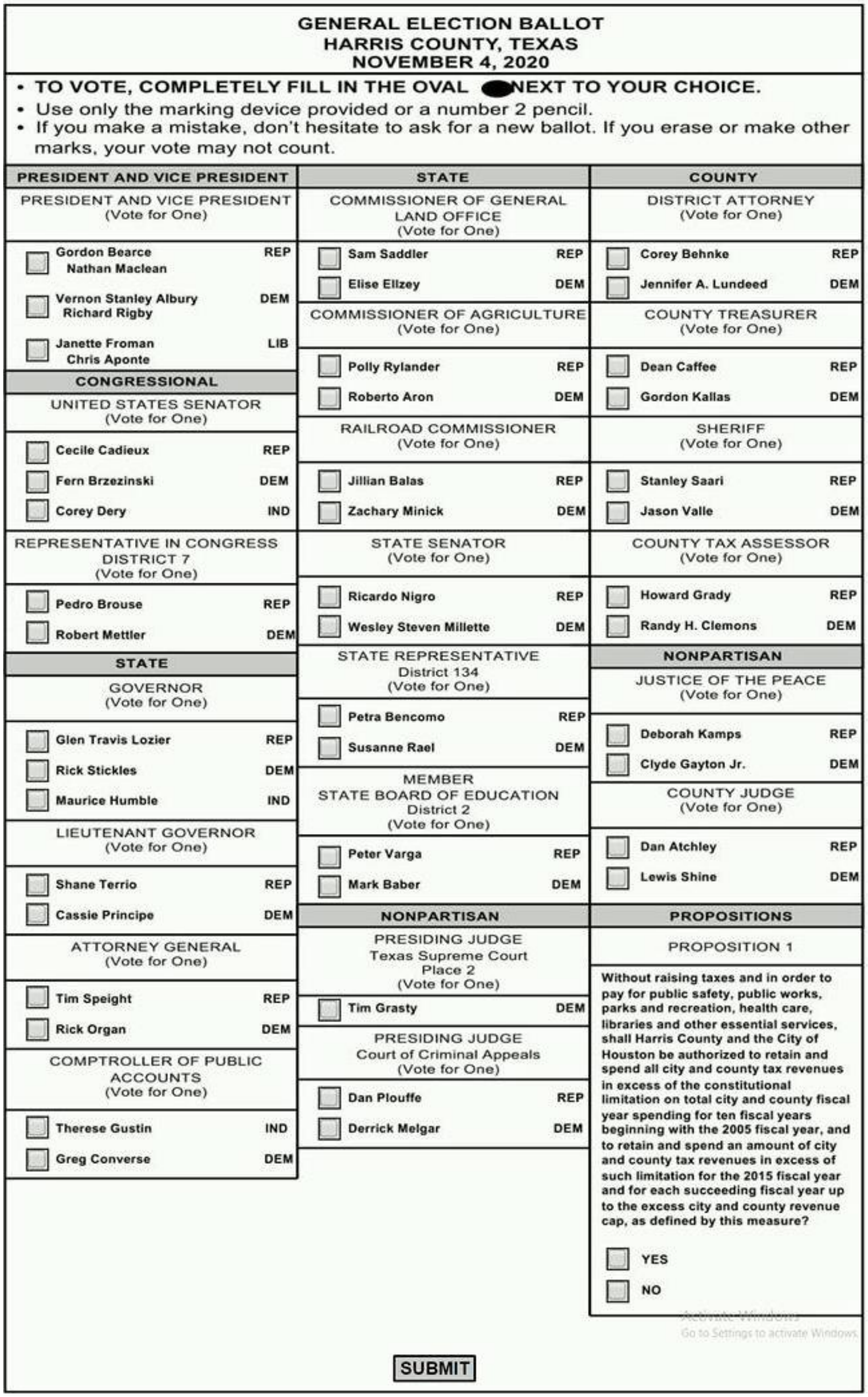
2.1 Method

2.1.1 Participants

A total of 28 (14 male, 14 female) Rice University undergraduate students were recruited. The ages ranged from 18 to 23 years, with an average of 19.6. The participants were compensated with credit toward a course requirement.

2.1.2 Material

The voting task mimicked the ballot that one would typically see at a voting booth if using a paper ballot: 21 races and one proposition were listed in a single display (see Figure 2.1). This was constructed by taking the picture of a paper ballot, making it the background of an HTML page, and adding checkboxes next to the candidates using CSS to simulate bubbling-in a vote.



GENERAL ELECTION BALLOT
HARRIS COUNTY, TEXAS
NOVEMBER 4, 2020

- TO VOTE, COMPLETELY FILL IN THE OVAL NEXT TO YOUR CHOICE.
- Use only the marking device provided or a number 2 pencil.
- If you make a mistake, don't hesitate to ask for a new ballot. If you erase or make other marks, your vote may not count.

PRESIDENT AND VICE PRESIDENT

PRESIDENT AND VICE PRESIDENT (Vote for One)

- Gordon Bearce / Nathan Maclean — REP
- Vernon Stanley Albury / Richard Rigby — DEM
- Janette Froman / Chris Aponte — LIB

CONGRESSIONAL

UNITED STATES SENATOR (Vote for One)

- Cecile Cadieux — REP
- Fern Brzezinski — DEM
- Corey Dery — IND

REPRESENTATIVE IN CONGRESS DISTRICT 7 (Vote for One)

- Pedro Brouse — REP
- Robert Mettler — DEM

STATE

GOVERNOR (Vote for One)

- Glen Travis Lozier — REP
- Rick Stickles — DEM
- Maurice Humble — IND

LIEUTENANT GOVERNOR (Vote for One)

- Shane Terrio — REP
- Cassie Principe — DEM

ATTORNEY GENERAL (Vote for One)

- Tim Speight — REP
- Rick Organ — DEM

COMPTROLLER OF PUBLIC ACCOUNTS (Vote for One)

- Therese Gustin — IND
- Greg Converse — DEM

STATE

COMMISSIONER OF GENERAL LAND OFFICE (Vote for One)

- Sam Saddler — REP
- Elise Ellzey — DEM

COMMISSIONER OF AGRICULTURE (Vote for One)

- Polly Rylander — REP
- Roberto Aron — DEM

RAILROAD COMMISSIONER (Vote for One)

- Jillian Balas — REP
- Zachary Minick — DEM

STATE SENATOR (Vote for One)

- Ricardo Nigro — REP
- Wesley Steven Millette — DEM

STATE REPRESENTATIVE District 134 (Vote for One)

- Petra Bencomo — REP
- Susanne Rael — DEM

MEMBER STATE BOARD OF EDUCATION District 2 (Vote for One)

- Peter Varga — REP
- Mark Baber — DEM

NONPARTISAN

PRESIDING JUDGE Texas Supreme Court Place 2 (Vote for One)

- Tim Grasty — DEM

PRESIDING JUDGE Court of Criminal Appeals (Vote for One)

- Dan Plouffe — REP
- Derrick Melgar — DEM

COUNTY

DISTRICT ATTORNEY (Vote for One)

- Corey Behnke — REP
- Jennifer A. Lundeed — DEM

COUNTY TREASURER (Vote for One)

- Dean Caffee — REP
- Gordon Kallas — DEM

SHERIFF (Vote for One)

- Stanley Saari — REP
- Jason Valle — DEM

COUNTY TAX ASSESSOR (Vote for One)

- Howard Grady — REP
- Randy H. Clemons — DEM

NONPARTISAN

JUSTICE OF THE PEACE (Vote for One)

- Deborah Kamps — REP
- Clyde Gayton Jr. — DEM

COUNTY JUDGE (Vote for One)

- Dan Atchley — REP
- Lewis Shine — DEM

PROPOSITIONS

PROPOSITION 1

Without raising taxes and in order to pay for public safety, public works, parks and recreation, health care, libraries and other essential services, shall Harris County and the City of Houston be authorized to retain and spend all city and county tax revenues in excess of the constitutional limitation on total city and county fiscal year spending for ten fiscal years beginning with the 2005 fiscal year, and to retain and spend an amount of city and county tax revenues in excess of such limitation for the 2015 fiscal year and for each succeeding fiscal year up to the excess city and county revenue cap, as defined by this measure?

- YES
- NO

SUBMIT

Figure 2.1 : The multi-race ballot interface that mimicked the ballot one would typically see at a voting booth. 21 races and one proposition question were listed.

To simulate the paper ballot experience, a 22-inch HP monitor was rotated vertically so that the ballot could fill the entire monitor without the need to scroll through it. A Gazepoint GP3 eye-tracker and Gazepoint Analysis UX Edition software were used to record the eye-tracking fixation data and fixation map videos. JSON was used to log interactions between the participants and the ballot: each JSON file contained information on the clicks, including what type of object was clicked (screen, voting for a candidate, unvoting for a candidate, submission) and its dissected components (race, political party, candidate name); what time the participant made each click; and the subject ID and date of the experiment. The JSON log also included the time the ballot was first opened, the time the ballot was submitted, and the duration of the entire voting process.

2.1.3 Procedure

At the beginning of the experimental session, the participants were given instructions on how to select candidates. They were randomly assigned either a voter guide that listed each candidate's platform, thus allowing the participants to vote freely, or a voting slate instructing them to vote for specific individuals. The participants were then calibrated to the eye-tracker, with a minimum calibration score of six out of nine for both eyes. Then, the experimenter started the eye-tracking recording, and the participants were taken to the ballot interface, where they completed their voting. Once they were finished, the participants clicked the submit button, which prompted the download of the JSON log file, and the experimenter ended the recording.

2.2 Data Analysis

For data analysis, both the participants' reading patterns and their fixation data associated with three ROIs—race title, candidates' names, and party affiliation—were examined, based on the JSON files, fixation data files, and fixation map videos. A challenge in the data analysis was that, for some participants, the fixation data was not properly aligned with their JSON logs. Adjustments to the fixation data therefore had to been made by carefully comparing the fixation data with the JSON logs and watching the fixation map videos.

However, not everyone had their fixation data and fixation map video perfectly recorded. Three out of the 28 participants had missing fixation map videos, and, of the reminder, 12 had only partial fixation data recorded for various reasons—some wore eyeglasses, some had jerky eye movements, and some held the voting instructions in front of their eyes throughout the task, which blocked the eye-tracker. As a result, 15 participants were excluded from the analysis of the ROIs.

2.3 Results

According to the fixation data and/or the JSON files, 20 participants voted with a strict serial *top to bottom left to right* pattern—that is, they started in the top-left corner and voted from the top to the bottom of that column, and then went over to the next column and went all the way to the bottom, repeating until they finished the voting task. Seven participants made minute adjustments but still followed the overall *top to bottom left to right* pattern most of the time. For example, one participant started voting with the *top to bottom left to right* pattern, read—but skipped—the race "Lieutenant Governor" (see Figure 2.1), then read and voted the two races below it, then came back and voted the “Lieutenant Governor”, and then finished the voting task with the *top to bottom left to right* pattern. As another example, there were five participants who jumped to the proposition race—located in the bottom-right corner—in the middle of their *top to bottom left to right* voting processes, voted for it, and then went back and picked up where they left off to finish the task. Most interestingly, given that the voting instructions provided were in a *top to bottom left to right* sequence, there was one participant who used a “snake” reading pattern (see Figure 2.2); it can also be noticed that the County Treasurer race was out by this participant.

Regarding the reading pattern, it was also found that nine participants skimmed the ballot and checked it before they hit the submit button to finish the task. In addition, four participants read through the instructions at the beginning of voting. These two numbers should be considered with caution due to the aforementioned fixation recording issues and the lack of corresponding JSON logs as backup.

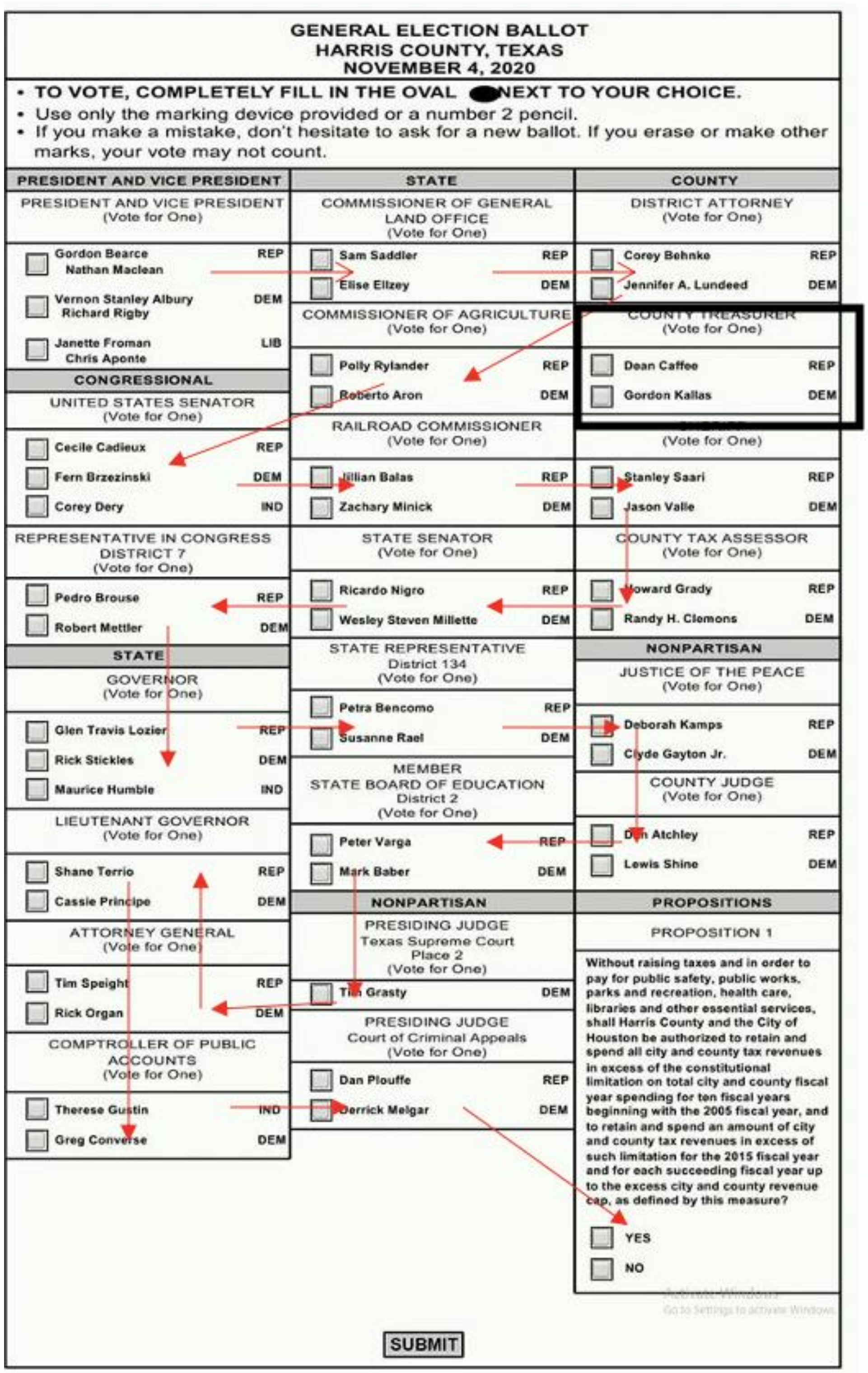


Figure 2.2 : One of the 28 participants used a “snake” reading pattern (indicated by red arrows), given that the voting instructions provided were in a top to bottom left to right sequence. “County Treasurer” was out by this participant.

Figure 2.3 shows summarized ROI proportions for 13 participants for all 22 races. Since a few participants had interesting interactions with the proposition question, the ROI use for the proposition question was studied independently. As can be seen in this figure, the participants interacted with the ballot quite differently—some showed great interest in the proposition question (s02, s09, s12, s23), and some participants rarely checked party affiliations (s11, s15, s20).

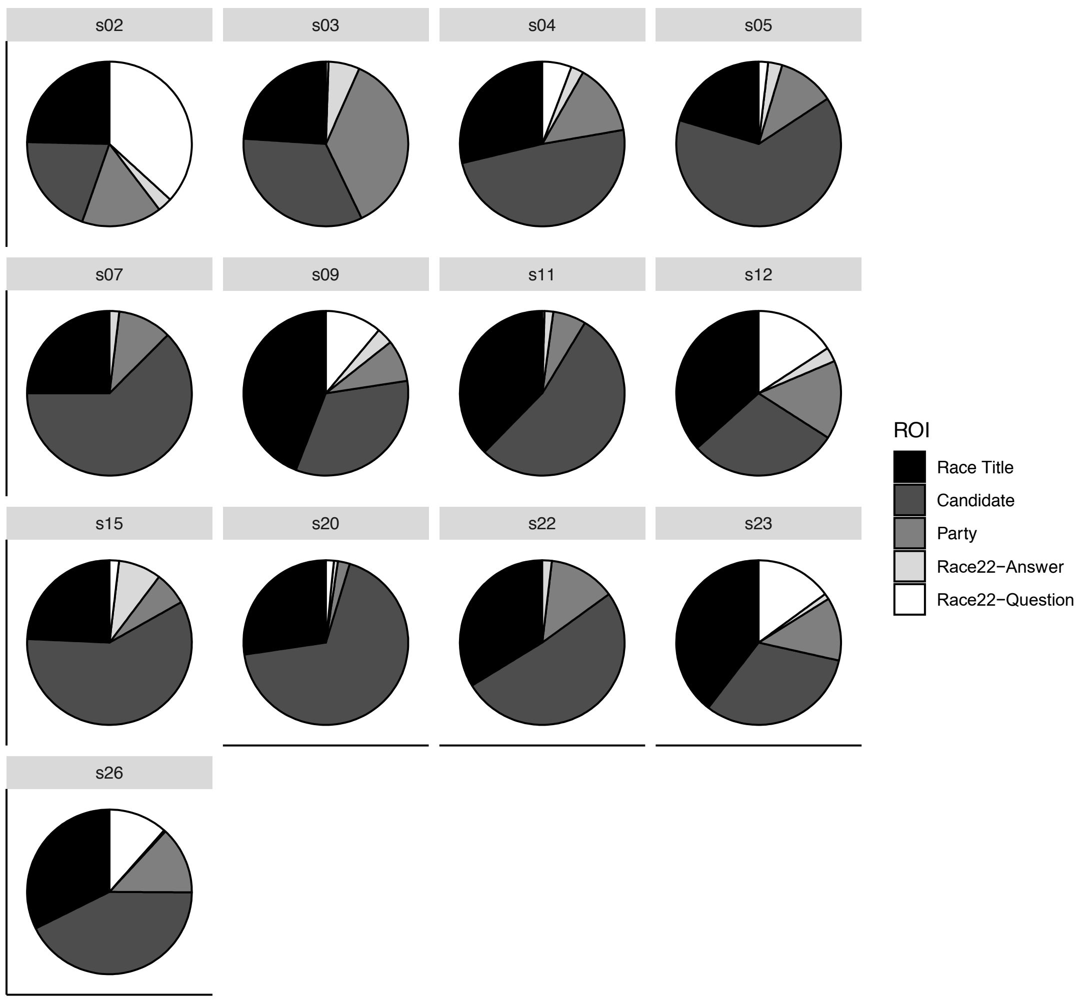


Figure 2.3 : ROI proportions for 13 participants for all 22 races. Participants showed different interest on three ROI areas: race title, name, party affiliation. The ROIs of the proposition question (race 22) was independently displayed.

To provide more information on the ROIs, Figure 2.4 displays the breakdown of each participant's ROI use for each race. The dynamic changes in the ROI proportions across the races can be observed—some participants appeared to use a more consistent set of ROIs (e.g., s11, s23), while others seemed to have different ROIs from race to race (e.g., s02, s03).

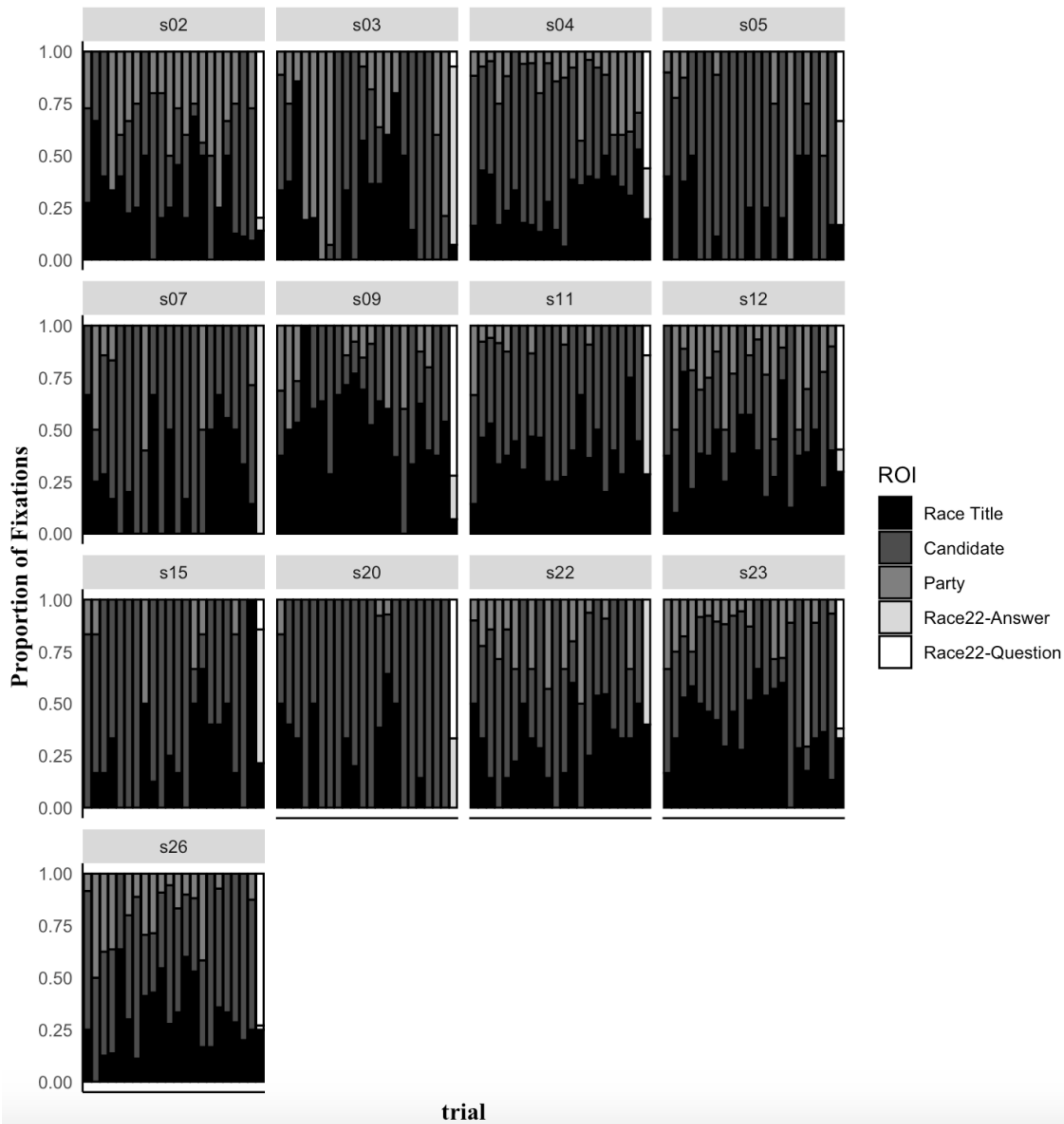


Figure 2.4 : ROI proportions (y-axis) for 22 races (x-axis, first trial on left) completed by 13 participants. Some participants favored a more consistent set of ROIs, while others had different ROIs from race to race. Missing data or non-ROI fixations are not presented in this figure.

2.4 Discussion

The results of Study 1 suggest two primary findings. First, *top to bottom left to right* was the most commonly used pattern for navigating between the races on the paper ballot. Second, the participants appear to have different reading patterns and switch memory strategies between races, which reinforces the importance of covering and modeling various voting strategies in error prediction. As can be seen in Figure 2.4, some participants (e.g., s03, s07, s20) read the race title in some races but not in others, implying that they used a recognition-based strategy occasionally, while for some participants (e.g., s04, s11), the race titles were consistently fixated, implying that a retrieval-based strategy was used throughout the task.

It is also noticeable that one of the 28 participants had the “snake” reading pattern (see Figure 2.2) and skipped one race. This finding demonstrates that at least a portion of voters may use non-traditional reading patterns in real elections and these patterns can be error-inducing. It highlights the importance of studying voter error using a thorough exploration of the strategy space to capture a wide array of errors.

Furthermore, although *top to bottom left to right* was observed as the most common reading pattern in this study, it is doubtful that the orderings of the races that were listed in the voting instructions might have influenced the reading patterns of the participants. Therefore, to investigate whether there is an effect of the race order, a future study will mix the race orders on the voting slates and voter guides, and then replicate the eye-tracking experiment.

Chapter 3

Study 2

The primary goal of Study 2 was to expand the strategy space covered by the multi-strategy model and to develop various plausible strategies for voting on full-face paper ballots. Four voting strategies, two visual search strategies, and five ballot knowledge levels were inherited from the multi-strategy model. Even though 40 combinations of strategies and ballot knowledge may seem a lot, they are not sufficient to encompass voters' behaviors with a full-face paper ballot. A full-face ballot, which contains many races on a single display, usually implies a larger number of plausible voting strategies and more sophisticated modeling work, as the model also needs to navigate from race to race. Several modeling issues therefore need to be addressed.

First, since more than one race is contained in a single display—and so the model also needs to navigate from race to race—macronavigation strategies were also considered and modeled in this study, in addition to the existing micronavigation strategies covered in the multi-strategy model. Macronavigation represents the process of moving from one race to the next, and micronavigation is the process of choosing the intended candidate to vote for within each race. Since the findings of Study 1 suggest that voters have different reading patterns throughout the voting task, two types of macronavigation pattern were modeled in this study.

Second, races are usually small and tightly arranged on a paper ballot. When there is significantly more information in a confined space, it is reasonable to believe that voters may also consider a race or even a column as a supergroup and consider race titles, candidates' names, and party affiliations as subgroups. It is therefore more complex for the model to identify race titles, candidates' names, and party affiliations; to distinguish different races; and to navigate from one race to another. In this study, the visual grouping algorithm was used (Lindstedt & Byrne, 2018), but, because its ability to group visual items is somewhat limited—in that the algorithm cannot identify and store supergroups nor link supergroups to the corresponding subgroups—an alternative solution had to be figured out to worked around this.

Besides developing a whole family of models, another goal of this study was to understand how errors emerge from the interaction of strategy and ballot design, given that tens of thousands of ballot designs are deployed for each national election. To achieve this goal, a series of randomly generated ballots was used to test the expanded multi-strategy model developed in this study.

3.1 Method

3.1.1 Ballot Design

The first step of the study was to create ACT-R compatible full-face ballots as the voting tasks. The voting tasks for the model consist of a virtual screen populated with several columns of races. The design of the ballots is clear and simple, with each race in its own clear visual group: each race has a title, a list of candidates and their associated parties, and a list of buttons that the model can click to vote for a candidate. In addition, on these ballots, the races in each column are horizontally aligned, as might be expected.

As mentioned earlier, the visual grouping algorithm can simulate how most voters group things within a race, but, for a paper ballot voting task, the groups generated by the grouping algorithm might only be considered as subgroups. To make the model compatible with the visual grouping algorithm and thus able to navigate between races, two types of ballot design were developed.

For the first type of ballot, the race headers were colored red, the candidates were colored purple, and the parties were colored blue (see Figure 3.1). The coloring allows the model to make visual location requests, like "the closest red text in the column to the right" (when finding the closest race) or "the closest purple text to my current position" (when finding the candidate group of the currently attended race). Since humans can normally reliably differentiate between race headers, candidates, and parties by using the visual characteristics of the ballot, it is believed that coloring the ballot does not give the model an unfair advantage. However, an alternative method was also explored to work around this problem.

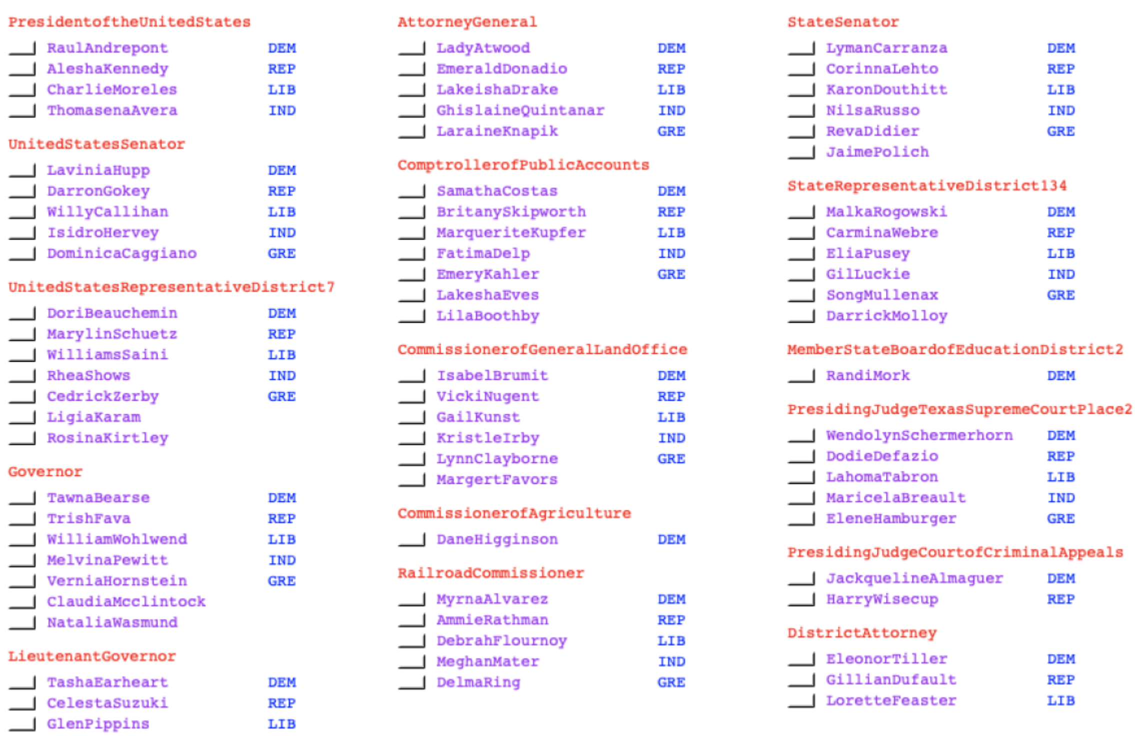


Figure 3.1 : A full-face ballot with colored texts. The coloring allows the model to make visual location requests to navigate between and within races.

The alternative ballot was built on the basis of the first type of ballot: besides the coloring cues used in the first method, an image background was added for each race (see Figure 3.2), and thus every subgroup layered on that image would be associated to the same race. The background allows the model to make visual location requests, like "the closest image in the column to the right" (when finding the closest race) or "the closest red text within the boundaries of the current image" (when finding the race group of the current race).

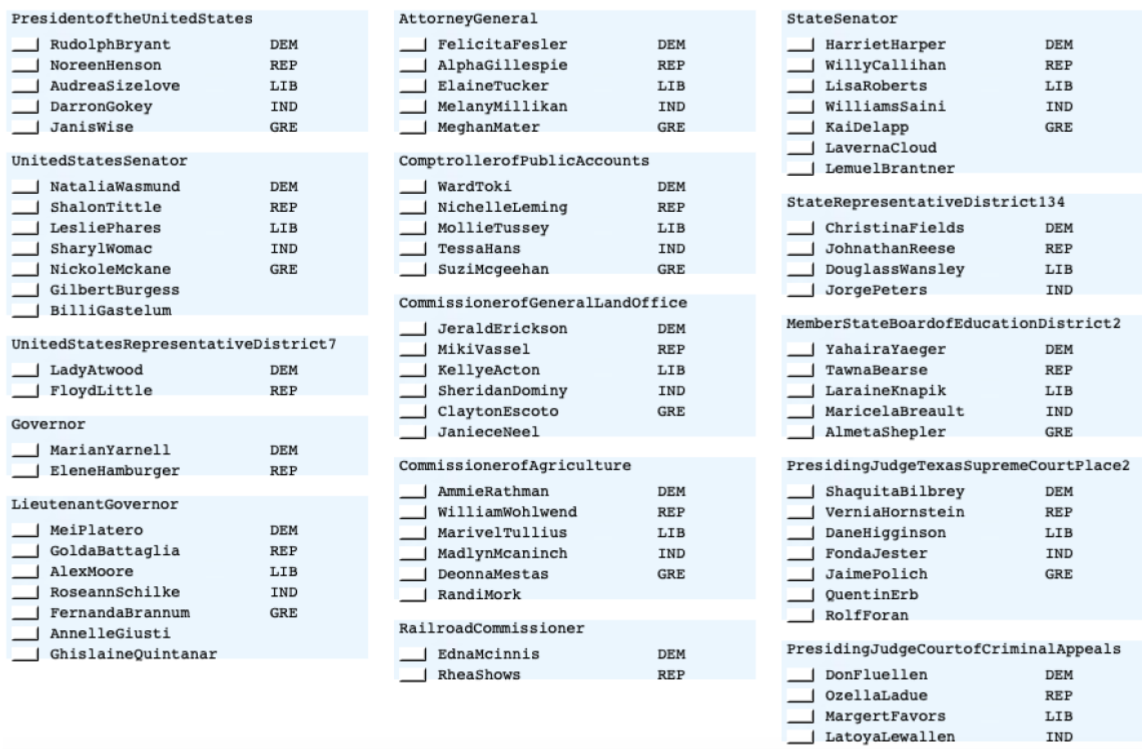


Figure 3.2 : A full-face ballot with image backgrounds. Race title, candidate names, and party affiliations that layered on a single image would be associated with each other. The image background allows the model to make visual location requests to navigate between races.

Since, on the first type of ballot, the model uses race titles to navigate between races and the race titles are at the top of each race, this type of ballot is described as *top-based*, and the model that uses race titles to navigate is also described as *top-based*. On the second type of ballot, the model uses images to navigate between races: the model first finds the center of the current image/race and then navigates to the center of another image/race for which it wants to vote. This type of ballot is therefore called *center-based*, and the model that uses images to navigate between races is also called *center-based*.

Furthermore, in order to understand the interaction between voting strategy and ballot design, both types of ballot were not static. Instead of consisting of a manually-positioned set of races and candidates, the ballots can be dynamically generated throughout the simulation processes. As each ballot was generated, each race was randomly selected to have several candidates and then placed a set distance below the last race. Also, three layout variables were allowed to vary: the vertical space between the races, the vertical space between the race header and the candidates, and the vertical space between the candidates, which yields 132 possible combinations of spacing variables (see Table 3.1). The ranges of the variables that were chosen resulted in ballots that the model could still realistically parse but were nevertheless visually distinct.

Variable range (pixels)	Range (pixels)
Space between races	5–15
Space between header and candidates	20–22
Space between candidates	15–18

Table 3.1 : Three ballot layout variables.

3.1.2 Strategy Space

The strategy space covers a total of 160 voting models, and each model includes a macronavigation strategy, a level of ballot knowledge, and a micronavigation strategy.

Multiple plausible alternatives were used for each component: four macronavigation strategies, eight micronavigation strategies, and five levels of ballot knowledge.

3.1.2.1 Macronavigation Strategy

The macronavigation strategy indicates the strategy for navigating between races. There are many plausible macronavigation strategies that voters could use—some may read in a serial, race-by-race pattern from one corner to its diagonal opposite; some may prefer to randomly pick a race to vote; some may scan globally and read all the bold, large, or colored headers first; and some may be unable to retrieve all the candidates they intend to vote for at one time and may therefore have to read the ballot a second time to fill out what they left incomplete.

Four macronavigation strategies were developed in this study: *top to bottom left to right* and *left to right top to bottom* patterns, interacting with *top-based* and *center-based* strategies. With the *top to bottom left to right* strategy, the model starts from the top-left corner and finishes the columns one by one, from left to right. Similarly, with the *left to right top to bottom* strategy, the model starts with the upper-leftmost race on the ballot, then proceeds to the right, navigating to the closest race to the last race it voted in the next column over and repeating until it votes on a race in the last column. It then goes back to the beginning of the row, finds the next race down in the left column, and repeats voting from left to right. The model continues until it runs out of new races in the left column.

3.1.2.2 Micronavigation Strategy

The micronavigation strategy represents the strategy for choosing a candidate to vote for within each race. It covers the interactions between four memory strategies (retrieve-party, recognize-party, retrieve-recognize-party, party only) and two visual search strategies (serial, random) for casting a vote within each race. The four memory strategies and two visual search strategies were inherited from the multi-strategy model.

3.1.2.3 Ballot Knowledge

Ballot knowledge defines voters' level of knowledge of the races and candidates. Five levels of ballot knowledge were inherited from the multi-strategy model (see Table 1.1).

3.2 Model Evaluation

As the micronavigation strategies and ballot knowledge remain the same as in the multi-strategy model, the research focus here was on the macronavigation strategies and understanding how voting errors changed as a function of the ballot layout. To allow the model to vote, ALL-PERFECT was selected for the ballot knowledge and random, recognize-party was chosen for the micronavigation strategy throughout the evaluation process.

3.2.1 Top to Bottom Left to Right Macronavigation Strategy

For each of the 132 combinations of spacing variables (see Table 3.1), the model was tested on 20 randomly generated *center-based* ballots and 20 randomly generated *top-based* ballots. As each ballot was generated, each race was randomly selected to have between one and seven candidates. For each run, the exact race positions, the race order on the ballot, and the order in which the model voted on the races (including any races the model missed) were recorded. Since the ballot layouts are simple and clear and the *top to bottom left to right* strategy is the most obvious method of macronavigation, the model did not miss any race throughout the voting processes.

3.2.2 Left to Right Top to Bottom Macronavigation Strategy

3.2.2.1 Method

For each of the 132 possible combinations of spacing variables, the model was tested on 50 randomly generated *top-based* ballots (Engels et al., 2020). As each ballot was generated, each race was randomly selected to have between one and four candidates. Thus, the model was run on 6,600 ballots for a total of 158,338 individual races. Similarly, the exact race positions, the race order on the ballot, and the order in which the model voted on the races were recorded for each model run.

3.2.2.2 Results

An error happens when the model skips a race. As introduced earlier, the races in each column are horizontally aligned on the ballots. However, when the race lengths are allowed to vary, the races in different columns are not vertically aligned, as the generation process always placed each race a set distance below the last race. Since the macronavigation strategy proceeded from left to right, in cases where the races were vertically misaligned, the model could make errors. Note that when the ballot is a perfect grid where all races are vertically aligned, the model does not make errors. It is therefore the interaction of this strategy with the design of the ballot that results in errors.

Figure 3.3 shows an example of the model missing a race on a typical *top-based* ballot. As can be seen in this figure, when the model reaches the third race down in the left column (“United States Representative District 7”) it votes on that race and then proceeds along the row, selecting and voting on the closest race and repeating until it reaches the last column. The model then returns to the race at the beginning of the row and proceeds to the first race on the next row down (“Governor”). Here is where it makes its mistake: because the “Railroad Commissioner” race is the closest race to “Governor,” the model votes on "Railroad Commissioner" for its second race in the row and so skips "Commissioner of Agriculture." It never returns to vote on this race.

Figure 3.3 : The *top-based* model skips "Commissioner of Agriculture." The green arrows mark a *left to right top to bottom* macronavigation voting pattern.

Overall, the model's global error rate is around 13.04%, meaning that, on average, given a random race on a *top-based* ballot, there is a 13.04% chance that the model will not vote on it. This rate is certainly much higher than any experimental rate in human voters, but, as this strategy is nonstandard, this is to be expected. Of course, most people do not make anywhere near these many errors, but average error rates in the wild likely stem from outliers, such as this strategy.

3.2.2.2.1 Effects of Race Location

The first thing examined was the relationship of race location on the ballot to model error. As can be seen in Figure 3.4, there is a general trend of increasing errors across columns. In other words, races in columns that are further to the right are more likely to be skipped.

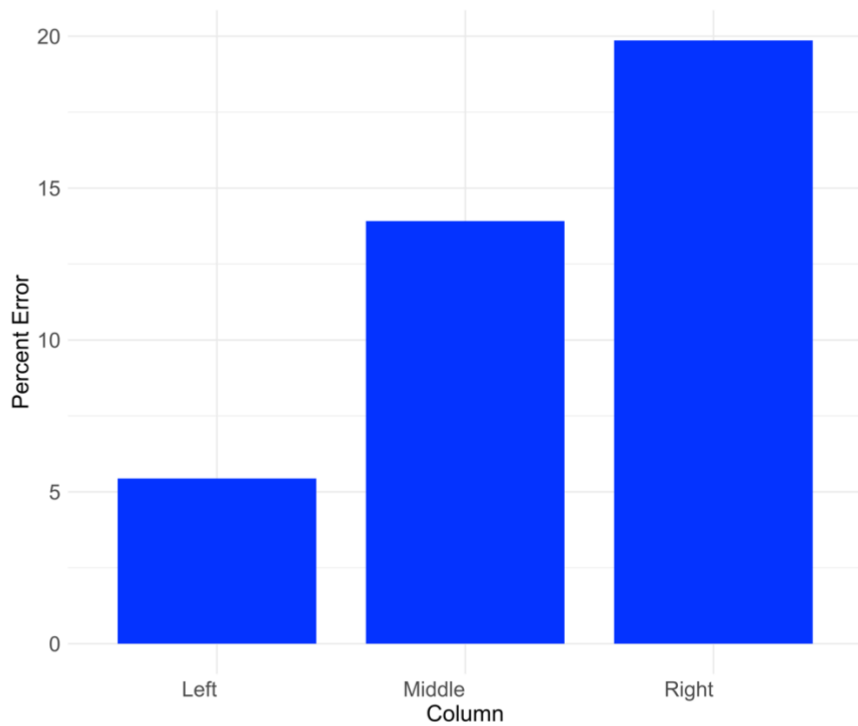


Figure 3.4 : Average voting error rate increased across races in the left, middle, and right columns across all ballot runs.

In fact, since the exact *y* coordinate and column for every race on every ballot were recorded, it is possible to generate a heatmap of error rates by race position on the ballot (see Figure 3.5). Each bin collates the percent error of the model for races within 10 vertical pixels, where the *y* position of a race is its header.

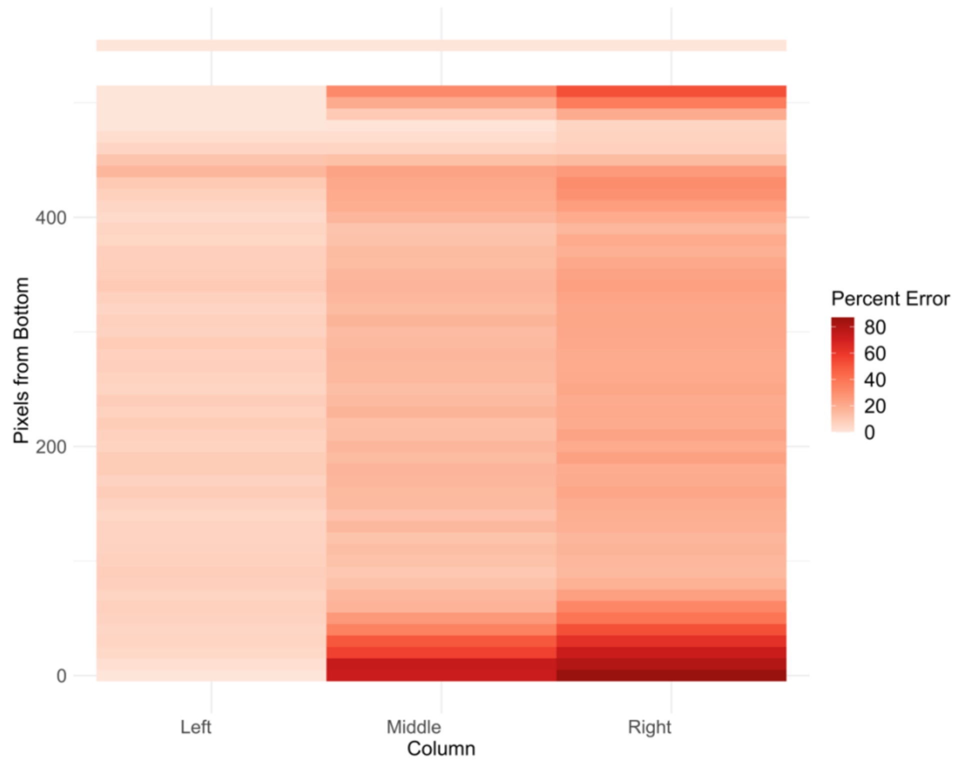


Figure 3.5 : Heatmap of the model's average voting error according to races' columns and *y*-axis positions.

Of interest are the places in Figure 3.5 where errors are likely. One immediately obvious place is the bottom-right corner, where average percent error approaches 100. The model almost always misses races here. To make sense of this result, it is observed that the only way in which a race can have its start in one of those bottom-right boxes is if it is very short. It makes sense that, for short races nestled in the bottom corner, people will frequently get to the last race in the left column and vote across that row not low enough to reach the bottom corner races.

However, other than this, errors are more or less uniformly distributed across the ballot. This result hints at the strength of the model: errors occur seemingly randomly across the ballot because they are emerging from the specific structure of individual random ballots.

3.2.2.2.2 Effects of Ballot Structure

Besides the race location, how specific elements of ballot structure influence the model error was also examined. First, differences in error rates with different amounts of vertical space between the end of each race and the beginning of the next were identified: voting error increased as the space between races decreased (see Figure 3.6). This result validates the intuition that the more cluttered a ballot is, the more likely the model is to miss a race.

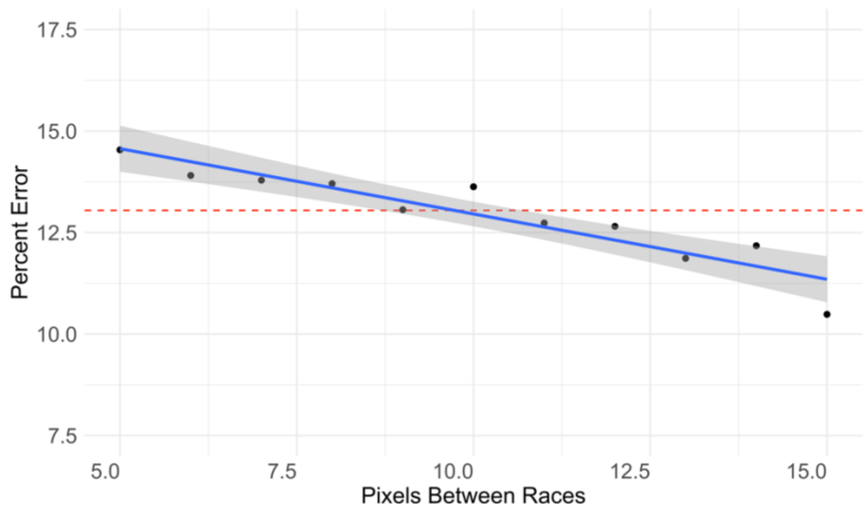


Figure 3.6 : Voting error increased as the space between races decreased. Each black dot is the average percent error across all ballots with a specific race spacing. The blue line is the linear regression for the trend, the red line is the average error of the model, and the shading represents the 95% confidence intervals for the line.

Recall that vertical space is just one of the spacing variables that was manipulated. Each specific vertical spacing value therefore includes many observations from ballots built from combinations of the other spacing variables. While these other spacing variables were also examined, no significant effect of them on the model's error rate was found.

The relationship between the length of a race and the chance it would be skipped was also been investigated and a similar result was found: as the length of a race decreased, the model's chance of skipping it (its error rate for races of that length) increased (see Figure 3.7). Of note, single-candidate races are most likely to be missed, but of course skipping such a race will not change the outcome of an election, since unopposed candidates are guaranteed to win.

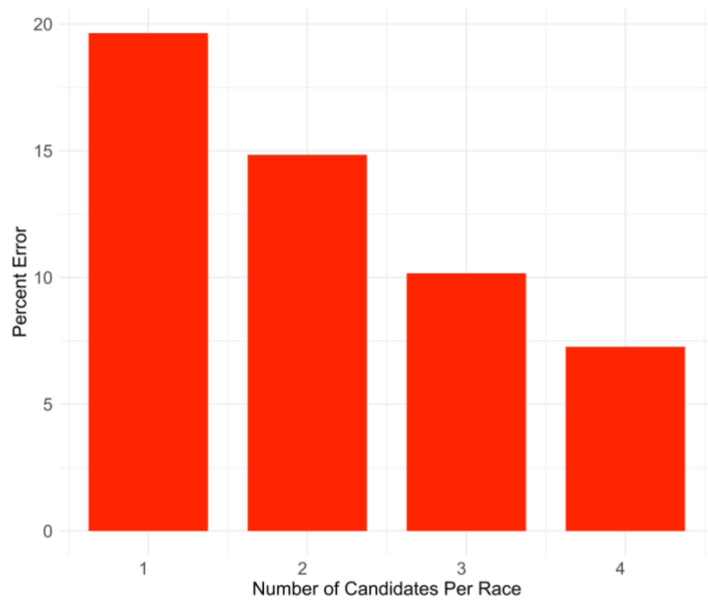


Figure 3.7 : Average error rate increased as the length of a race decreased.

Finally, how the model's error rate varied as a function of the vertical distance from a given race to the nearest race to it in the previous column was studied. Figure 3.8 shows a stacked bar plot of races missed and races voted on according to this variable. This graph shows two things: first, that the chance a simulated voter misses a race increases as the closest distance to the last race increases, and second, that the number of races that are far from any prior race decreases as the distance increases. The reason that the distribution is non uniform, with peaks in the 0 bin, 15–20 bin, 30–35 bin, and 45–50 bin, is a result of how the ballots were generated. The candidate spacing varied from 15 to 18 pixels, and it was frequently the case that the closest race in the last column was an integer multiple of candidate space away (see Figure 3.3).

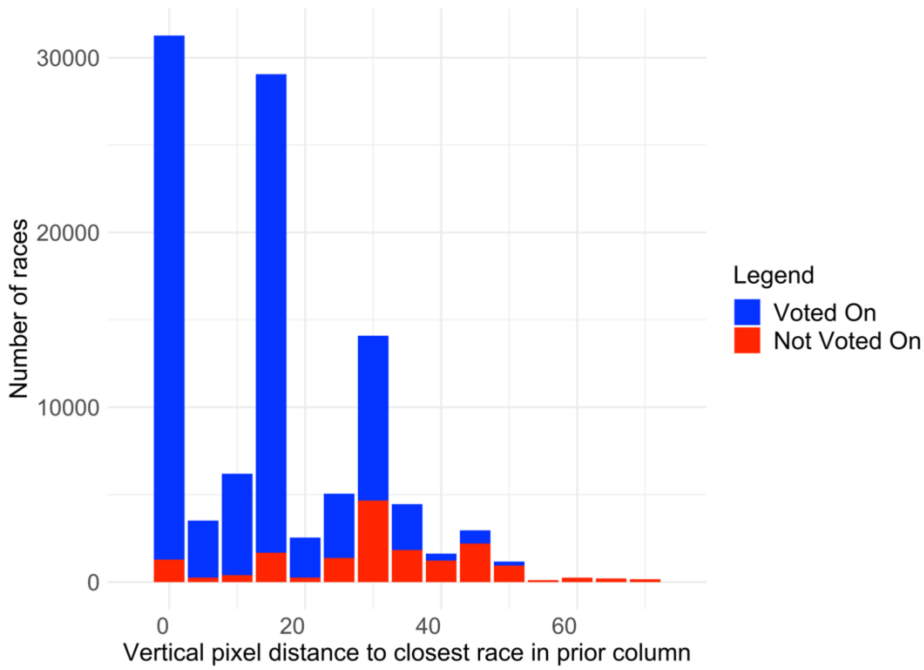


Figure 3.8 : Stacked bar plot of the number of races voted on and not voted on across all model runs, plotted according to the vertical distance between the race and the closest race in the previous column (bins of 5 pixels).

3.2.2.3 Replication of the Findings

For each of the 132 combinations of spacing variables, the *center-based* model was also tested using 20 randomly generated ballots. As each ballot was generated, each race was randomly selected to have between one and seven candidates. The effects of both race location and effects of ballot structure were replicated (see Figure 3.9, Figure 3.10, Figure 3.11, Figure 3.12, Figure 3.13), but with slightly higher global error rates: on average, given a random race on a *center-based* ballot, there is a 16.58% chance that the model will miss the vote, compared to the 13.04% chance on a *top-based* ballot.

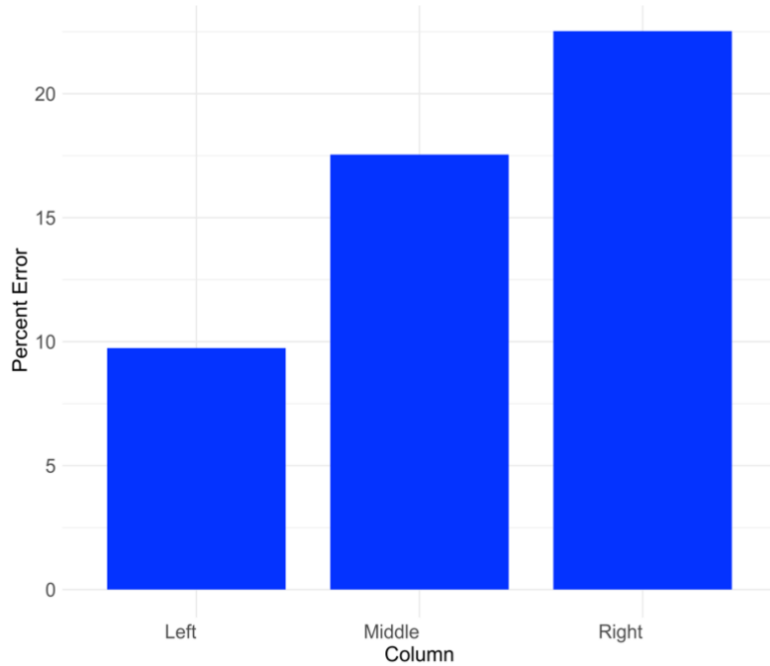


Figure 3.9 : Average voting errors across races in the left, middle, and right columns across all ballot runs. Replicated by *center-based* model.

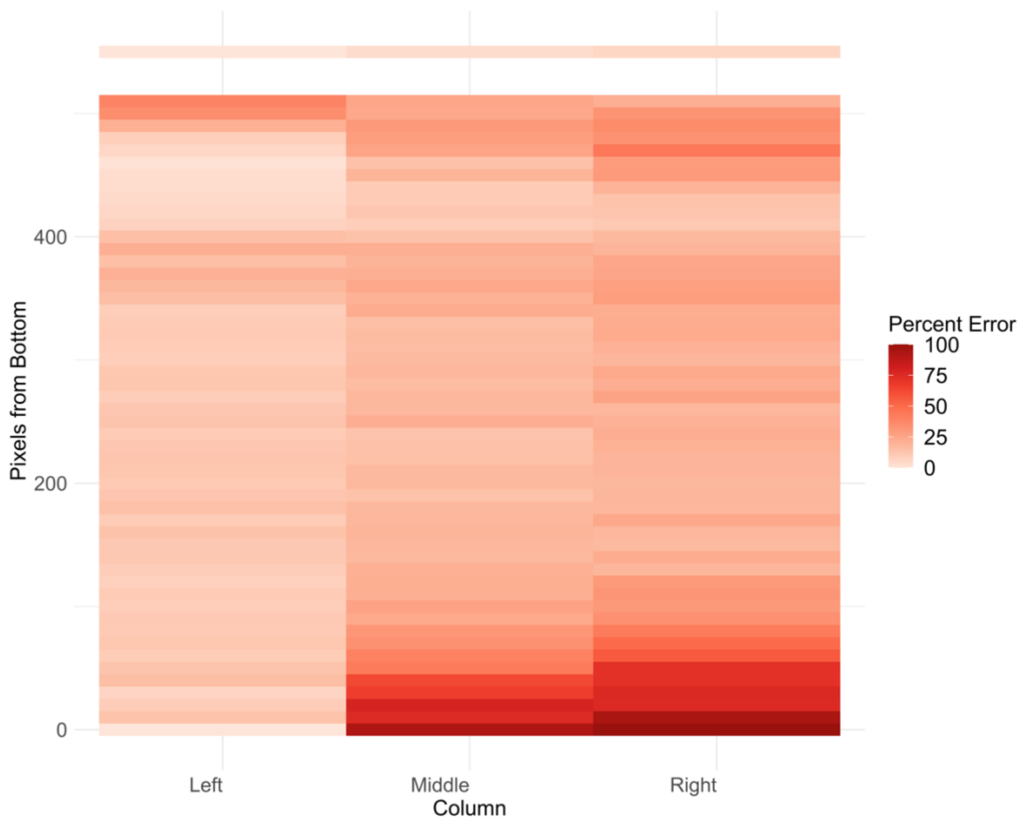


Figure 3.10 : Heatmap of the model's average voting error according to races' columns and *y*-axis positions. Replicated by *center-based* model.

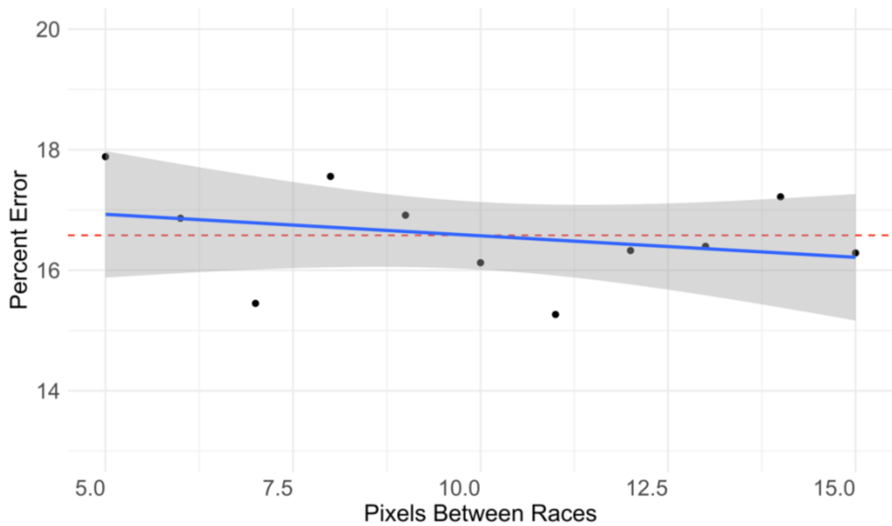


Figure 3.11 : Voting error increased as the space between races decreased. Replicated by *center-based* model.

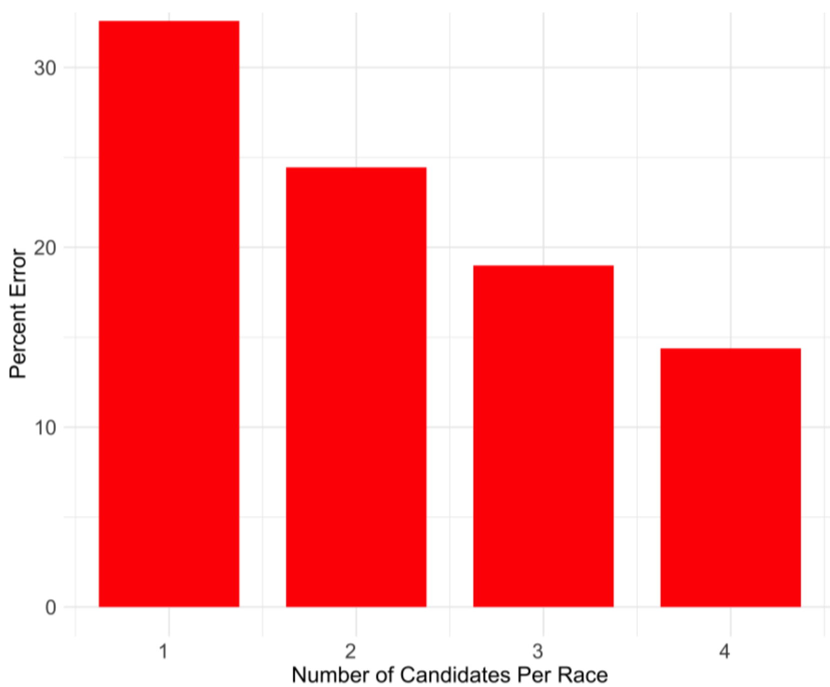


Figure 3.12 : Average error rate increased as the length of a race decreased. Replicated by *center-based* model.

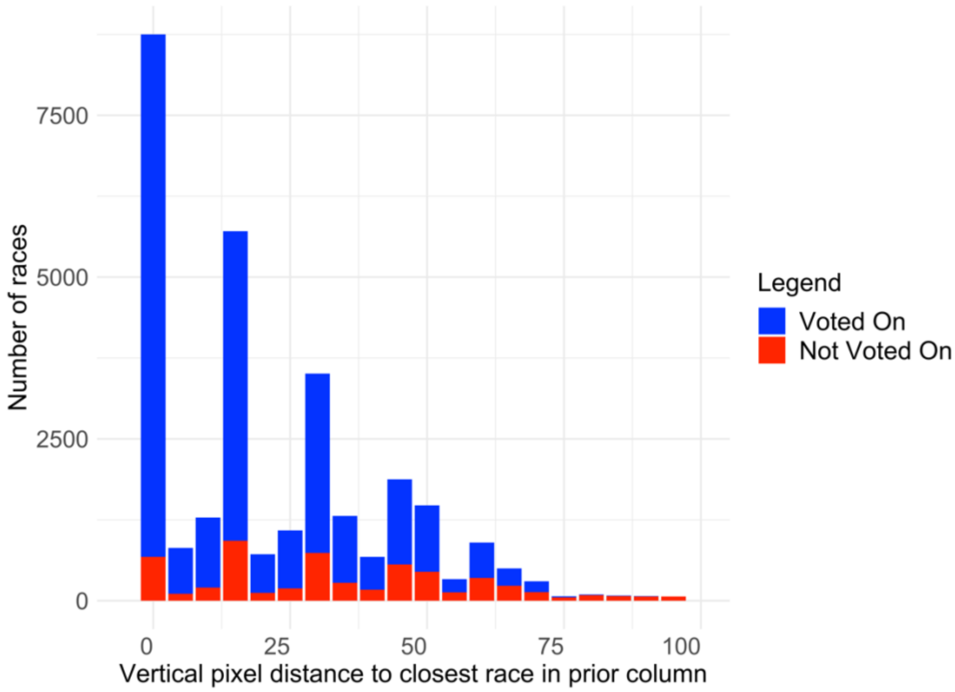


Figure 3.13 : Stacked bar plot of the number of races voted on and not voted on across all model runs. Replicated by *center-based* model.

The reason for the higher error rates is that the model uses different reference points to navigate between races on the two types of ballot: the *top-based* model uses race titles to find the locations of the next race to vote for, but, the *center-based* model navigates between races according to the center of images—the race titles are closer to the tops of the races, so a few missing races can be prevented. As an example, in Figure 3.14, the *center-based* model missed the “Attorney General” race, because the center of “Comptroller of Public Accounts” race was closer to the center of the “United States Senator” race. In contrast, this error could be avoided by the *top-based* model because the title “Attorney General” might be closer to the race title “United States Senator.”

Figure 3.14 : The *center-based* model voting using the *left to right top to bottom* macronavigation strategy. The model skips "Attorney General."

3.2.3 Conclusion

Races were more likely to be missed if they were smaller, out of alignment with the races in other columns, or more cramped overall. These are all characteristics of bad ballots that the model detected organically. The detection behavior emerged out of the design of the strategy; it was not hardcoded. The fact that the model's error behavior was unplanned and emergent is in line with the long-term plan of building models that can produce novel errors on novel ballots.

Indeed, it can be seen that the average error for this strategy is far higher than the average error for all voters, even assuming, as the model did, that once a voter finds a race, they would successfully vote on it (choosing a perfect micronavigation strategy, in the parlance of the model). Most real voters probably use a more successful macronavigation strategy. However, if even a subset of voters uses this strategy, or one like it, then it is necessary to account for them in the model, as a subset of voters can still have a deciding impact on a close race.

3.3 Model Validation

3.3.1 Method

For model validation, the model was tested on an infamous bad ballot: the ballot used in Wisconsin in 2002 (see Figure 1.2). Many voters made errors on this ballot because they considered two sections of the gubernatorial race as two different races. Since the model makes use of the visual grouping algorithm, it is important to ensure that the model has the ability to produce the “Wisconsin error.”

First, a simplified Wisconsin ballot was developed as the voting task (see Figure 3.15): the gubernatorial race on the Wisconsin ballot, which yielded overvotes, was reproduced and placed in the same location; for the rest of the ballot, races developed for previous voting tasks were inserted.

Figure 3.15 : The Wisconsin error was replicated on the simplified Wisconsin ballot. The gubernatorial race was split across two columns. The model identified the two sections as two races and voted in both sections.

The next issue to address is the number of Monte Carlo replications. The overall error rate generated by the model was expected to be around 5% and the 95% confidence intervals for the model predictions to be no wider than 5% in either direction. The table in Byrne (2013) shows this requires 109 model runs; 200 runs per model were therefore performed to be slightly more conservative.

3.3.2 Results

A total of 80 voting models, created from the interactions of two *center-based* macronavigation strategies, five levels of ballot knowledge, and eight micronavigation strategies, were tested on the ballot. The Wisconsin error was reproduced successfully: overall, the model generated an average 5.80% Wisconsin error rate across all voting models, which means that the model had a 5.80% chance of making a vote for both sections of the gubernatorial race.

Differences in error rates between macronavigation strategies and between visual search strategies were not found, which means that the choice between a *left to right top to bottom* or a *top to bottom left to right* strategy to navigate between races did not affect the results, nor did the choice between a serial or a random scanning pattern to navigate within races. However, differences in error rates were observed for memory strategies. As can be seen in Figure 3.16, there was clearly an effect of memory strategies on the Wisconsin error: about 10% more such errors were generated with the "recognize-party" strategy, but for the other three memory strategies, a difference in the error rates was not apparent. Note that the errors only occurred in the FULL-MEMORY condition; since, with FULL-MEMORY, the models could remember all candidates' names, it is suspected that the model could thus be capable of voting a second time in the second section of the gubernatorial race.

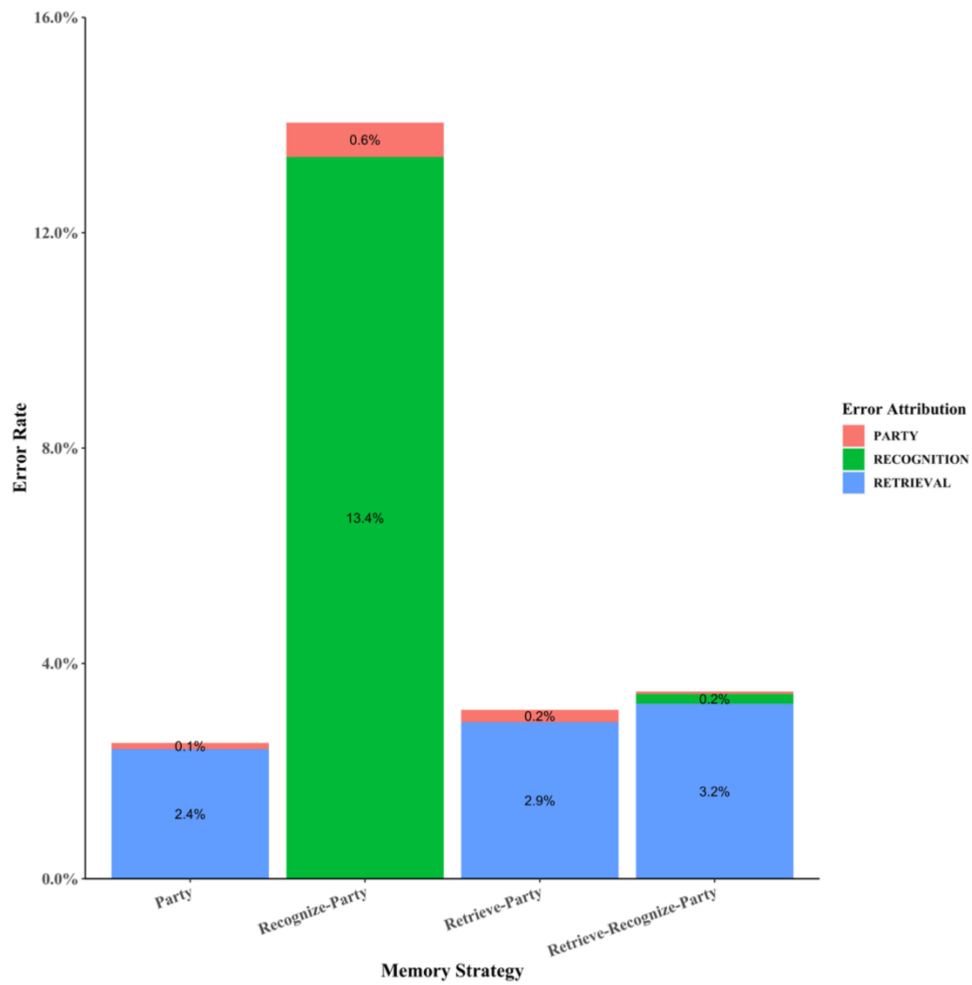


Figure 3.16 : Average Wisconsin error rate across four memory strategies with FULL-MEMORY.

3.4 Discussion

In Study 2, the research focus is on developing macronavigation strategies and understanding the interaction of macronavigation strategy and ballot design. The insight from Study 1 was successfully addressed and integrated: in addition to the traditional *top to bottom left to right* strategy, the model also makes use of a non-standard *left to right top to bottom* macronavigation strategy. Notably, using a non-standard macronavigation strategy amplified the ability to detect bad ballots. For instance, a strategy moving in the same direction as that in which the races were originally placed might not mind if the races were very close together, but any other strategy would. Ballot designers need to cater to less common strategies, so an ability to detect when ballots will cause systematic errors in voters using these strategies is crucial.

Through the model evaluation process, the effects of race locations and effects of ballot structures were identified. While some of the findings may seem obvious, they must partly be viewed in the light of the wider project. The model was able to vote on a wide array of ballots that looked visually different and to successfully make consistent errors. More than just characterizing the type of ballots and races that are more disposed to be skipped by a specific voter, these findings confirm the feasibility of attempting to eventually predict errors in novel ballots.

Furthermore, the model makes an interesting additional prediction: as it is more likely to miss races in the center and right columns and is more likely to miss smaller races, the model predicts that average voter error should be higher on down-ballot races in the real world (as some voters may use a similar left to right strategy). This skew is likely to be more severe in years with a presidential race, since there are often many candidates running for president, meaning that the first race in the left column would be very long, thus making it more likely that other columns' races will not be aligned.

Of course, the model has yet to be perfected, and there is still work to be done. First, the simulation is not the same as an actual paper ballot—filling out a ballot with a pencil is not the same as clicking a bubble with a mouse, and how to model the click actions and mouse noise must be carefully considered and implemented. Also, the voting tasks do not quite look like real full-face paper ballot—elements like instructions and the lines separating the races need to be added to give the voting tasks a more realistic look.

Second, although the model has successfully reproduced the Wisconsin error, it cannot yet deal with instructions, which means that the model cannot make errors that induced by the interactions of strategies and instructions (e.g., the ballot used in Broward County, Florida in 2018; see Figure 1.3). The model should be further updated so that its validity can be fully guaranteed.

Third, one must remember that the goal of this study is to simulate and model how people vote on paper ballots, and there are still voter behaviors not covered by the model. For example, some voters may fill in the wrong bubble and change it later; some may use other macronavigation strategies, such as the “snake” pattern (see Figure 2.2); and some may glance back over the ballot at their filled bubbles to check they filled everything out. All of these processes may introduce new sources of error. Therefore, it is necessary to keep exploring the strategy space and to model more voting strategies.

Chapter 4

Conclusion and Future Directions

This thesis represents an important step toward the end goal of constructing an automated error prediction tool to identify bad ballots: the model used a total of 160 different voting strategies constructed from differing memory and navigational strategy selections. More importantly, it represents the first use of ACT-R as an error prediction tool to diagnose whether there are particular combinations of strategies and ballot layouts that lead to voting errors.

Study 2 confirms the feasibility of predicting errors in paper ballots and the validity of the model—how ballot layout and the visual task strategy can interact to produce voting errors was systematically studied. The results can even be used to generate applied advice for a hypothetical election official who must build a ballot with races of varying length. Such an official should strive to line up race headers as much as possible, sacrificing races per page by leaving blank space so that races can be aligned (this would help increase accuracy not only with the specific *left to the right top to bottom* macronavigation strategy tested in this study, but indeed any strategy that goes left to right). Moreover, the official should try not to squeeze races into the bottom right corner, and in general try to keep the ballot uncluttered by putting as much space between races as possible. Figure 4.1 shows an example of bad ballot design. As can be seen on this ballot, races are vertically misaligned, with varying race lengths and limited spaces between races. It is therefore harder to distinguish different races comparing with voting on a properly designed ballot that prevent errors (see Figure 4.2). The official might even consider making the space within races more cramped to make the delineations between races clearer, although this will introduce the possibility for a voter filling in the wrong bubble or missing the candidate they want to vote for. Future models will be developed to predict these errors.

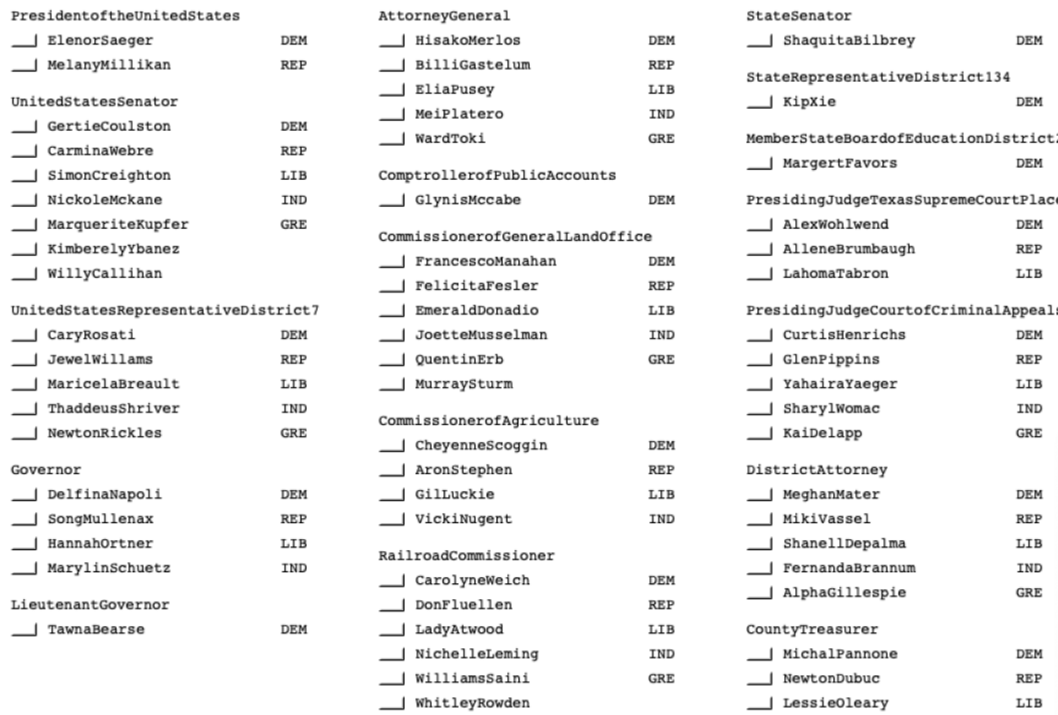

PresidentoftheUnitedStates
ElenorSaeger DEM
MelanyMillikan REP
UnitedStatesSenator
GertieCoulston DEM
CarminaWebre REP
SimonCreighton LIB
NickoleMckane IND
MarqueriteKupfer GRE
KimberelyYbanez
WillyCallihan
UnitedStatesRepresentativeDistrict7
CaryRosati DEM
JewelWillams REP
MaricelaBreault LIB
ThaddeusShriver IND
NewtonRickles GRE
Governor
DelfinaNapoli DEM
SongMullenax REP
HannahOrtner LIB
MarylinSchuetz IND
LieutenantGovernor
TawnaBearse DEM
AttorneyGeneral
HisakoMerlos DEM
BilliGastelum REP
EliaPusey LIB
MeiPlatero IND
WardToki GRE
ComptrollerofPublicAccounts
GlynisMccabe DEM
CommissionerofGeneralLandOffice
FrancescoManahan DEM
FelicitaFesler REP
EmeraldDonadio LIB
JoetteMusselman IND
QuentinErb GRE
MurraySturm
CommissionerofAgriculture
CheyenneScoggin DEM
AronStephen REP
GilLuckie LIB
VickiNugent IND
RailroadCommissioner
CarolyneWeich DEM
DonFluellen REP
LadyAtwood LIB
NichelleLeming IND
WilliamsSaini GRE
WhitleyRowden
StateSenator
ShaquitaBilbrey DEM
StateRepresentativeDistrict134
KipXie DEM
MemberStateBoardofEducationDistrict2
MargertFavors DEM
PresidingJudgeTexasSupremeCourtPlace
AlexWohlwend DEM
AlleneBrumbaugh REP
LahomaTabron LIB
PresidingJudgeCourtofCriminalAppeals
CurtisHenrichs DEM
GlenPippins REP
YahairaYaeger LIB
SharylWomac IND
KaiDelapp GRE
DistrictAttorney
MeghanMater DEM
MikiVassel REP
ShanellDepalma LIB
FernandaBrannum IND
AlphaGillespie GRE
CountyTreasurer
MichalPannone DEM
NewtonDubuc REP
LessieOleary LIB

Figure 4.1 : A poorly designed ballot.

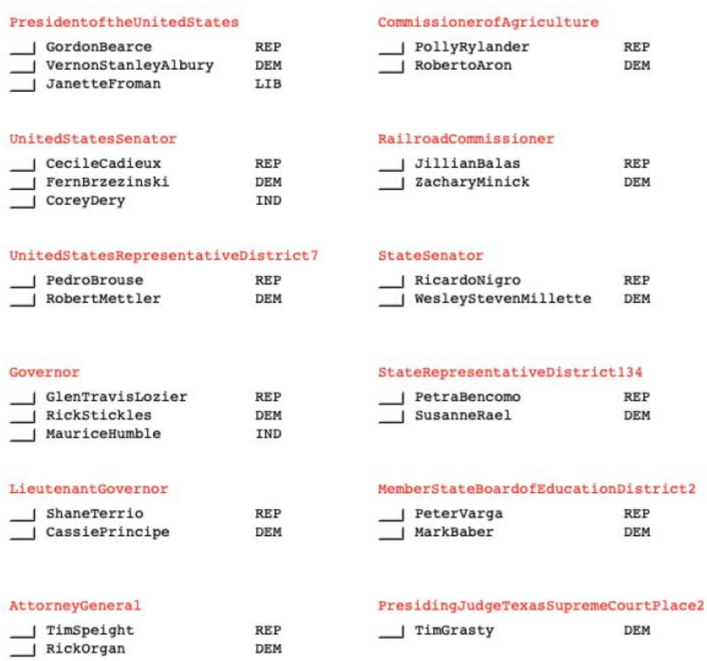

PresidentoftheUnitedStates
GordonBearce REP
VernonStanleyAlbury DEM
JanetteFroman LIB
UnitedStatesSenator
CecileCadieux REP
FernBrzezinski DEM
CoreyDery IND
UnitedStatesRepresentativeDistrict7
PedroBrouse REP
RobertMettler DEM
Governor
GlenTravisLozier REP
RickStickles DEM
MauriceHumble IND
LieutenantGovernor
ShaneTerrio REP
CassiePrincipe DEM
AttorneyGeneral
TimSpeight REP
RickOrgan DEM
CommissionerofAgriculture
PollyRylander REP
RobertoAron DEM
RailroadCommissioner
JillianBalas REP
ZacharyMinick DEM
StateSenator
RicardoNigro REP
WesleyStevenMillette DEM
StateRepresentativeDistrict134
PetraBencomo REP
SusanneRael DEM
MemberStateBoardofEducationDistrict2
PeterVarga REP
MarkBaber DEM
PresidingJudgeTexasSupremeCourtPlace2
TimGrasty DEM
CountyTreasurer
DeanCaffee REP
GordonKallas DEM
Sheriff
StanleySaari REP
JasonValle DEM
CountyTaxAssessor
HowardGrady REP
RandyHClemons DEM
JusticeofthePeace
DeborahKamps REP
ClydeGaytonJr DEM
CountyJudge
DanAtchley REP
LewisShine DEM

Figure 4.2 : A simple and clear ballot layout.

One of the next steps will be to completely map the space of macronavigation strategies by running eye-tracking experiments on human subjects voting on different ballot layouts and by studying and integrating the insights from the analysis results. To implement these new strategies, it is also inevitable that capabilities of ACT-R itself will be expanded by extending the current visual grouping module to group objects in a hierarchy and by extending the options that models must visually navigate.

New sub-strategies for other parts of the model are also planned, including new ways for the model to encode the candidate, party, and race groups and to find and click the bubble corresponding to a candidate. Again, a group of simulations will need to be conducted to determine every behavior variant. A robust automatic ballot usability evaluation system that can dynamically build any voter from the voting strategy space will thereby be developed, and a wider variety of errors will be capture and prevented. Most importantly, more diverse ballots can be tested before deployment.

References

Aaltonen, A., Hyrskykari, A., & Räihä, K.-J. (1998). 101 spots, or how do users read menus? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '98*, 132–139. https://doi.org/10.1145/274644.274664

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* (pp. ix, 290). Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195324259.001.0001

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, *38*(4), 341–380. https://doi.org/10.1006/jmla.1997.2553

Annett, J., & Duncan, K.D. (1967). Task analysis and training design. *Journal of Occupational Psychology 41*, 211–221.

Appel, A. W. (2018, November 14). Florida is the Florida of ballot-design mistakes. *Freedom to Tinker.* https://freedom-to-tinker.com/2018/11/14/ florida-is-the-florida-of-ballot-design-mistakes/

Botvinick, M., & Plaut, D. C. (2004). Doing without schema hierarchies: a recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review*, *111*(2), 395–429.

Byrne, M. D. (2013). How many times should a stochastic model be run? An approach based on confidence intervals. *Proceedings of the 12th International Conference on Cognitive Modeling*, (pp. 445–450). Ottawa, Canada: Carleton University.

Cooper, R. P., & Shallice, T. (2006). Hierarchical schemas and goals in the control of sequential behavior. *Psychological Review, 113*(4), 887–916.

Engels, J., Wang, X., & Byrne, M. D. (2020). Missed one! How ballot layout and visual task strategy can interact to produce voting errors. *Proceedings of the 18th International Conference on Cognitive Modeling*.

Everett, S. P. (2007). *The usability of electronic voting machines and how votes can be changed without detection* [Doctoral dissertation, Rice University]. Rice University Electronic Theses and Dissertations. https://scholarship.rice.edu/handle/1911/20601

Everett, S. P., Greene, K. K., Byrne, M. D., Wallach, D. S., Derr, K., Sandler, D., & Torous, T. (2008). Electronic voting machines versus traditional methods: Improved preference, similar performance. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 883–892. https://doi.org/10.1145/1357054.1357195

Fleetwood, M., & Byrne, M. D. (2006). Modeling the visual search of displays: A revised ACT-R model of icon search based on eye-tracking data. *Human-Computer Interaction*, *21*(2), 153–197. https://doi.org/10.1207/s15327051hci2102_1

Goldstein, D. G., & Gigerenzer, G. (2002). Models of ecological rationality: The recognition heuristic. *Psychological Review*, *109*(1), 75–90. https://doi.org/10.1037/0033-295X.109.1.75

Greene, K. K. (2010). *Effects of multiple races and header highlighting on undervotes in the 2006 Sarasota general election: A usability study and cognitive modeling assessment* [Doctoral dissertation, Rice University]. Rice University Electronic Theses and Dissertations. https://scholarship.rice.edu/handle/1911/70254

Halbrügge, M., Quade, M. & Engelbrecht, K.-P. (2015). A predictive model of human error based on user interface development models and a cognitive architecture. In

Taatgen, N. A., van Vugt, M. K., Borst, J. P., & Mehlhorn, K. (Eds.), *Proceedings of the 13th International Conference on Cognitive Modeling*, (pp. 238–243). Groningen, the Netherlands: University of Groningen.

Laskowski, S. J., Autry, M., Cugini, J., Killam, W., & Yen, J. (2004). *Improving the usability and accessibility of voting systems and products* (NIST SP 500-256; 0 ed.). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.500-256

Lebière, C., Anderson, J. R., & Reder, L. M. (1994). Error modeling in the ACT-R production system. In A. Ram, & K. Eiselt (Eds.), *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, (pp. 555–559). Hillsdale, NJ: Erlbaum. https://doi.org/10.4324/9781315789354-96

Lindstedt, J. K., & Byrne, M. D. (2018). Simple agglomerative visual grouping for ACT-R. In I. Juvina, J. Houpt, & C. Myers (Eds.), *Proceedings of the 16th International Conference on Cognitive Modeling*, (pp. 68–73). Madison, WI: University of Wisconsin.

Lindstedt, J. K., Byrne, M. D., Li, Y., & Wisdom, H, M. (2019). *The deceptively simple task of casting a vote: Detecting unanticipated memory strategies using eye tracking* [Unpublished manuscript, Rice University].

Miller, J. M., & Krosnick, J. A. (1998). The impact of candidate name order on election outcomes. *Public Opinion Quarterly*, *62*(3), 291–330. https://doi.org/10.1086/297848

Schooler, L. J., & Anderson, J. R. (1997). The role of process in the rational analysis of memory. *Cognitive Psychology*, *32*(3), 219–250. https://doi.org/10.1006/cogp.1997.0652

Norden, L. D., Kimball, D. C., Quesenbery, W., & Chen, M. (2008). *Better Ballots*. Brennan Center for Justice at New York University School of Law.

Sandler, D., Derr, K., & Wallach, D. S. (2008). VoteBox: A tamper-evident, verifiable electronic voting system. *Proceedings of the 17th Conference on Security Symposium*, 349–364.

Trafton, J. G., Altmann, E. M., & Ratwani, R. M. (2011). A memory for goals model of sequence errors. *Cognitive Systems Research*, *12*(2), 134–143. https://doi.org/10.1016/j.cogsys.2010.07.010

Wand, J. N., Shotts, K. W., Sekhon, J. S., Mebane, W. R., Herron, M. C., & Brady, H. E. (2001). The butterfly did it: The aberrant vote for Buchanan in Palm Beach County, Florida. *American Political Science Review*, *95*(4), 793–810. https://doi.org/10.1017/S000305540040002X

Wang, X., Lindstedt, J. K., & Byrne, M. D. (2019). The model that knew too much: The interaction between strategy and memory as a source of voting error. In Stewart, T.C. (Ed.), *Proceedings of the 17th International Conference on Cognitive Modeling*, (pp. 283–288). Waterloo, Canada: University of Waterloo.