Computational Models of Regressive Eye Movements in Reading

by

Craig A. Sanders

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of Bachelor of Science

With Honors in Brain, Behavior, and Cognitive Science

From the University of Michigan

2013

Advisor: Richard Lewis

Abstract

Several computational models of eye movements in reading have been developed over the years, but none can agree on the cause of *regressions*, movements to previously read words. Some models regard regressions as errors or exceptions to more general rules about eye-movements, but this study proposes a model which explores the idea that regressions are governed by the same mechanism as forward saccades, that regression behavior may be influenced by task goals, and that regressions are an emergent phenomenon in an optimal reading strategy. Human participants were given a reading task in one of three conditions that differentially rewarded speed and accuracy, and their results were compared to those of a series of computer simulations which found the optimal reading strategy for each condition using the proposed model. Although the model's behavior did not match the humans' in all respects, some important properties were captured, and the behavioral study did confirm the prediction that regression behavior is affected by task goals. Furthermore, the model's failings provide insight into the human behavior and open pathways for future research.

Keywords: reading, eye-movements, computational modeling, regressions, eye-tracking

Computational Models of Regressive Eye Movements in Reading

Between novels, emails, text messages, and even road signs, most people today read in some capacity every single day, and it is hard to imagine how society could function without this skill. However, it is often overlooked how complex and cognitively demanding this task actually is. Reading requires the precise coordination of eye movements, visual attention, and linguistic processing, and the suppression of both external and internal distractions.

Many researchers in psycholinguistics have attempted to build computational models to explain how these processes work, but this task has proved difficult because the underlying mechanisms governing reading are mostly unconscious and opaque to the reader. For instance, readers of English are often unaware that although most of their eye movements go from left-to-right, roughly 10% of their saccades are actually right-to-left movements called *regressions* (Reichle, Pollatsek, Fisher, & Rayner, 1998). Regressions are hard to induce experimentally, so little is known about what causes them (Rayner, 1998) but being able to explain regressions is important for any model of reading for two reasons. Firstly, answering why readers make regressions answers a more fundamental question: how do readers choose the next word to read? Secondly, it has been shown that models which allow for regressions are able to read more quickly and accurately than models that do not allow for regressions (Bicknell & Levy 2010a), so they must be a part of any good reading strategy. This point is especially important for models which explore how rational agents should behave.

In this paper I will provide an overview of several computational models of reading, with a particular emphasis on what these models have to say about regressions. Then I will outline a new model which acts as an ideal observer with some imposed, physiological constraints. This model attempts to find the optimal eye-movement strategy for a reading task that is performed under varying payoff conditions which differentially reward accuracy and speed. I will then assess whether the model makes regressions in a manner similar to humans. If it does, that may imply that the model and the humans have a similar underlying mechanism for making regressions.

Overview of Present Research

E-Z Reader

E-Z Reader tries to answer *when* and *where* people move both their eyes and their attention as they read. It belongs to the class of *cognitive-control* models of reading because it posits a tight link between the mind and the eyes. The goal is not to merely understand reading but to understand how cognition interacts with perception and motor control (Reichle, Pollatsek, Fisher, & Rayner 1998).

When E-Z Reader was first formulated, it only had two core assumptions. The first is that attention is required for lexical access and is allocated serially to one word at a time; the second is that saccadic programming is decoupled from shifts of attention. Lexical access occurs in two stages: *L1* and *L2*. The authors remain somewhat agnostic on the psychological interpretations of these stages and choose to focus more on how they function in the model, but *L1* is often referred to as the "familiarity check" and *L2* is referred to as the "completion of lexical access." *L1* begins on word *n* once attention has been allocated to it, and when *L1* is completed a saccade is programmed to word *n* + 1, the word immediately to the right. While the saccade is being programmed, *L2* begins on word *n*. Once *L2* is complete, word *n* has been identified, and attention is then moved to word *n* + 1. If word *n* + 1 is highly predictable, *L1* may occur instantaneously, in which case a saccade to word *n* + 2 will be programmed while the eye is still fixated on word *n*, meaning that word $n + 1$ will be skipped (Reichle, Pollatsek, & Rayner, 2006).

Because it is known that regressions are more prevalent in difficult text (Jacobson & Dodwell, 1979; Rayner & Pollatsek, 1998) and regressions often occur after errors in comprehension (Blanchard & Iran-Nejad, 1987; Frazier & Rayner, 1982), early versions of E-Z Reader assumed that regressions were caused by errors in higher level linguistic processes and thus were out of the scope of the model (Reichle et al., 1998). However, later versions of the model have some additional assumptions that allow for regressions. E-Z Reader 10 introduces a post-lexical integration stage of processing after *L2*. This stage has some probability of failing, and when that occurs, a regressive saccade is made to the offending word (Reichle, Warren, & McConnell, 2009).

In sum, regressions in E-Z Reader are exceptions to a more general rule. The model always chooses the word to the immediate right of the current fixation as its target and only revisits previous words when an error has occurred.

SWIFT

SWIFT (Saccade-generation with inhibition by foveal targets) differs from E-Z Reader in that attention and lexical access are not applied to words serially. Instead, lexical access is performed simultaneously on all words within an “attentional window” that extends from one word to the left of the current fixation to two words to the right. When attention is allocated to a word, its lexical activity increases until reaching some maximum value, at which point lexical access is complete, and lexical activity drops back down to zero. Another difference is that SWIFT decouples the decision of when to make a saccade from the decision of where to make a saccade. Unlike other models, SWIFT does not assume that saccade programming is modulated by high level linguistic processing. Instead, it assumes that saccades are controlled by a mostly autonomous, stochastic system that tries to maintain a constant mean rate of saccades (Engbert, Longtin, & Kliegle, 2002; Engbert, Nuthmann, Richter, & Kliegle, 2005). However, much research has shown that low frequency words produce longer fixation times than high frequency words (Rayner, 1998), and obviously such a “dumb” system would not be able to account for this effect. Therefore the probability of initiating a saccade may be inhibited by foveal lexical activity; when the fovea is fixated on a difficult word, it may suppress the probability of making a saccade. The decision of where to saccade is also modulated by lexical activity, with words with higher activity having a greater chance of being targeted than words with lower activity, but this processes occurs independently of saccade timing. This target selection mechanism is motivated by the idea that important target words are in an intermediate state of lexical processing, with their lexical activities close to their maximal amounts (Engbert, et al., 2002; Engbert, et al. 2005).

Two types of regressions can occur in SWIFT. Since the probability of making a saccade to a word is determined by that word’s lexical activation, any word within the attentional window with nonzero lexical activity can be targeted simply due to chance. When a saccade is made to the word immediately to the left of the current word, this is known as a “local” regression due to its limited length. The second type of regression, called a “global regression,” occurs when a word has not been completely processed before it has exited the attentional window. Once a word moves outside the attentional window, its degree of lexical activity is maintained, and so it remains a candidate for target selection, and its chance of being targeted increases over time, which reflects difficulties in comprehension (Engbert, Longtin, & Kliegle, 2002). Therefore, as in E-Z Reader, regressions in SWIFT are caused by difficulties in linguistic processing, but unlike in E-Z Reader, regressions can be explained using the same mechanism as forward saccades.

The Bicknell and Levy Model

Bicknell and Levy (2010b) have criticized SWIFT's mechanism for regressions as being irrational. Moving away from a word before it has been identified, only to return to it later, is illogical and inefficient, and this is contrary to a growing body of work of rational analysis (Anderson, 1990) in language processing (Hale, 2011) and eye-movement control (Legge, Klitz, & Tjan, 1997).

Bicknell and Levy claim the problem with both SWIFT and E-Z Reader is that they gloss over how word identification actually works, and they identify words with absolute certainty, but it has been shown that word identification is a noisy process and that readers maintain some uncertainty about the identities of words they have already read (Levy et al., 2009). Bicknell and Levy have presented their own model of reading which performs word recognition via Bayesian inference, and develops probabilistic beliefs about word identities, with high probabilities indicating confident beliefs and low probabilities indicating uncertain beliefs. Under this framework, occasionally making regressions actually is rational.

At each timestep, the model receives noisy information from several letters at a time, with the noise of each letter being a function of that letter's distance from the model's fovea. The model uses this information to update its belief about both the letters it is currently looking at and its belief about prior letters. For instance, take the case where the model only knows two strings, *AB* and *BA*, and only gets information from one letter at a time. After obtaining some noisy information about the first letter, the model believes that that letter is *A*, and moves on to the second letter. If the model gets information from the second letter that again indicates that it is *A*, the model's certainty about the first letter will decrease because there is no string *AA*.

The model can take one of three actions at each timestep: remain at the currently fixated position, stop reading, or make a saccade to a new position, with a two timestep delay representing the time it takes to program and execute a saccade. The model has two thresholds, α and β. If the belief about the current position is less than α, the model will remain fixated at that position. Otherwise, if the belief of some prior position is less than β, the model will saccade to the closest such position. If the belief of all positions up to the current position is greater than β, the model will saccade to *n* characters past the closest position to the right of the current fixation whose belief is less than α, where *n* is an arbitrary parameter used to ensure that the model moves forward at a decent pace. If no such position to the right exists, the model stops reading.

Bicknell and Levy found that not only are regressions rational in this model, but regressive policies are actually both faster and more accurate than otherwise equivalent non-regressive policies (i.e., policies where $\beta = 0$). The intuitive reason that regressive policies are more accurate is that regressions allow the model to reread words and correct for initially bad input. The reason that they are faster is that they allow for a lower value of α. When regressions are not allowed, a high value of α is desirable because the model only gets one chance to look at letters and thus needs to be highly confident about their identities before moving on. But when regressions are allowed a lower value of α is permissible because the model can go back and reread words if its uncertainty about those words drops later on. This means fewer timesteps will be needed to collect enough information to develop a high enough belief to move on to the right, thus increasing reading speed.

Although Bicknell and Levy's model is successful at explaining how regressions can be part of a rational reading strategy and corrects some of the problems of E-Z Reader and SWIFT, it does have its own limitations. Notably, it does not provide a comprehensive explanation for how forward saccades work.

Mr. Chips

Mr. Chips is an ideal observer model that reads text in the minimum number of saccades. As Mr. Chips reads, he constantly tries to identify the *current word*, or the leftmost word in the text that has not been identified. Mr. Chip often has some partial information about the current word such as its length or the identity of some of the letters. Mr. Chip finds the subset of words in its lexicon that is consistent with that partial information and makes a saccade according to the following rule: make the rightmost saccade that will, on average, minimize the uncertainty about the current word. Mr. Chips cannot leave the current word until it is identified unambiguously. Regressions are an emergent phenomenon under this simple rule. Making saccades far to the right is a bit of a gamble because some letters to the left might be left out of the visual field and remain unidentified. This could leave the leftmost word ambiguous, in which case a regressive saccade will be necessary to fill in the missing letters. Oftentimes, however, Mr. Chips will have enough information about the words to left to be able to infer what the missing letters are without actually looking at them, and therefore is able to skip some words and move through the text more quickly (Legge et al., 1997).

Although Mr. Chips was not originally formulated to directly model human behavior but rather to serve as a benchmark, it has been shown to produce eye movements similar to those of humans (Legge et al., 2002). Bicknell and Levy (2010b) have found fault with this model as well, however. They point out that on average Mr. Chips produces shorter regressions and skips fewer words than human readers. They claim that Mr. Chips makes two oversimplifications. The first is that, like E-Z Reader and SWIFT, Mr. Chips identifies words with complete certainty rather than developing probabilistic beliefs about word identities. The second is that Mr. Chips does not take linguistic context into account when performing word identification. Bicknell and Levy implemented two changes to the model which made its behavior correspond even more strongly to real human eye movements. First, the model moves from the current word not when it is identified unambiguously but rather when its belief about the word's identity crosses some threshold, and second, the model now takes the identities of prior words into account when identifying the current word.

By making risky saccades and occasionally allowing for regressions, Mr. Chips is able to read texts in fewer saccades on average than if he used a more conservative strategy in which all letters to the left must be viewed before any letters to the right. Therefore as in Bicknell and Levy's model, regressions in Mr. Chips are not aberrations, but necessary components of an ideal reading strategy, and it is able to explain regressions using the same mechanism as forward saccades without having to posit separate mechanisms for decisions about where and when to saccade. Furthermore, Mr. Chips posits an elegant solution for deciding where exactly forward saccades should land, unlike Bicknell and Levy's model. However, this model only accounts for saccade trajectories and says nothing about fixation durations.

The Bounded Optimal Model

The preceding models all have their own strengths and weaknesses, but one limitation that they all share is that they cannot account for the effect of task demands on eye movement strategies. Lewis, Shvartsman, and Singh (to appear) had participants perform a reading task in varying conditions that differentially awarded points for reading either quickly or accurately.

They showed that people unconsciously adapt to these varying task demands by making shorter fixations when they are required to read quickly and longer fixations when they are required to read accurately. Furthermore, they showed that the fixation times produced by participants in the different conditions aligned with those predicted by their computational model, which found the optimal reading strategy for each condition. This model will henceforth be referred to as the Bounded Optimal model because the driving idea behind it was bounded optimality, the notion that human behaviors are approximately optimal adaptations to the joint constraints of the human information processing system, the external probabilistic environment, and an internal reward function.

Currently this model cannot account for regressions because it only allows for forward saccades. But task goals may play an important role in regressions as well. It may be that people will make more regressions when they are required to read accurately because regressions would allow for a second chance at identifying words. It could also be, though, that regressions are more frequent when people are required to read quickly because reading too quickly may cause problems in comprehension, and as noted before, regressions are often associated with difficulties in linguistic processing. Whatever the case may be, a model that could account for such differences would be useful, so the Bounded Optimal model will provide the basis for the model that will be the focus for the rest of the paper.

The List Lexical Decision Task

In order to understand the Bounded Optimal Model, one must first understand the List Lexical Decision (LLDT), an extension of the Lexical Decision task first introduced by Meyer and Schvaneveldt (1971). On each trial of the LLDT, participants are presented with a list of 6 character strings consisting of 4 letters each. The goal is to report, via a button press, whether all strings are real English words or if at most one of the strings is a nonword. No strings are repeated in the same list.

Both human participants and the computational model are evaluated on the LLDT according to three payoff conditions: accuracy, balanced, or speed. In each condition, when participants get a trial correct they are awarded some points according to how quickly they responded, but they also lose points when they get a trial incorrect. The speed payoff awards fewer points than the other payoffs for the same reaction times, so participants are pressured more to respond quickly. In contrast, the accuracy payoff awards more points for the same reaction times, but it penalizes incorrect responses much more severely, so participants are encouraged to take their time in this condition. Complete quantitative descriptions of each payoff scheme can be found in Table 1.

The LLDT was chosen for a number of reasons. Firstly, it requires both the application of linguistic knowledge and the control of serial visual attention through eye movements, so it allows hypotheses about the interaction between high level and low level reading processes to be tested. Secondly, participants always start out fixated on the leftmost string, and the quickest means to get through the trial would be to simply read the strings in order. Therefore it should produce left-to-right reading and yield an eye-tracking record similar to natural reading. Finally, the task allows for trial-by-trial feedback and quantitative payoff schemes that differentially reward speed and accuracy.

Word Recognition via Bayesian Inference

Like Bicknell and Levy's model, one of the core assumptions of the Bounded Optimal model is that there is some noise in acquiring perceptual information from written words and in the process of matching orthography to the lexicon, and to overcome this noise, the model iteratively uses Bayesian inference. However, the Bounded Optimal model performs recognition at the level of whole words rather than individual letters as in Bicknell and Levy's model.

The model's representation of words is adapted from Norris' Bayesian Reader (2006). Each letter is represented as a vector of length 26 with a 1 in the position of the corresponding letter and a 0 in all other positions. Each string is comprised of four of such vectors. Samples are generated by adding mean-zero Gaussian noise to each element of each vector. The model only receives information from one word at a time, which is a reasonable assumption given the wide spacing of the strings in the human experiment (approximately 3.4 degrees of visual angle). This coding is not meant to represent how words and letters are actually stored in the human mind but rather provides a convenient means for performing mathematical operations, with the additional benefit of placing strings in a representational space with plausible similarity relations.

At each time step, the model receives a sample from the string it is currently fixated on and uses Bayes rule to update its belief about the string corresponding to each word in its lexicon and each nonword in its list of known nonwords. The priors over the words are derived from corpus frequencies from the Brown Corpus (Kucera & Francis, 1967). The model then updates its belief that there is a nonword in each position, taking into account the fact that there can be at most one nonword in every trial, but in order to reduce computation time, it does not take into account that strings cannot be repeated in each trial. The prior over a nonword being in each position is the probability of a nonword trial divided by the number of positions. The model then updates its belief that the current trial is a word trial (this belief this is the complement over the sum of there being a nonword in each position). The prior over the trial being a word trial is simply the probability of a word trial. For complete mathematical details, see Lewis et al.'s original article.

The Oculomotor Architecture

The model's perceptual inference mechanism is embedded within a larger oculomotor architecture, which is derived from current mathematical models of oculomotor control in reading. This architecture introduces some delays into the model. When the model fixates on a word, it does not obtain information from it right away, but after short lag due to the time it takes for information to travel from the eye to the brain (the eye-brain lag, VanRullen & Thorpe, 2001). Saccades and the motor actions required to press the response buttons also take time to program and execute. The means and standard deviations of these delays are reported in Table 2. The actual durations of these delays are drawn from gamma distributions, with the shape parameter $k = \frac{\mu^2}{\sigma^2}$ and the scale parameter $\Theta = \frac{\sigma^2}{\mu}$. The only parameters that were not set ahead of time were the sample rate and the noise parameter, which effectively serve as a scaling parameters by increasing or decreasing the time and the number of samples required to form a high belief probability that as string is either a word or nonword. The sample rate was fixed at one sample every 10 ms, and the noise parameter was fit to this. This fitting process is described in detail the method section.

The Old Policy Space

This section describes the Bounded Optimal Model as originally formulated by Lewis et al. At each time step, the model can take four different actions: program a saccade, program a motor response for "word trial," program a motor response for "nonword trial," or do nothing. The decisions to program a saccade and to program a motor response are not mutually exclusive. After a saccade or motor response has been programmed, it will automatically be executed after some delay (see preceding section) and cannot be cancelled. Regardless of what action the model takes, it will continue to receive samples from the string it is fixated on.

The model must decide when it has collected enough information from a string and should make a saccade to the next, and it must decide when it has collected enough information to decide what type of trial it is in and make a response. These decisions are governed by two thresholds: the saccade threshold and the decision threshold. As the model collects samples from a string, its belief about the string will perform a random walk towards either "word" or "nonword." When the model's belief that the string is either a word or a nonword crosses the saccade threshold, the model will program a saccade to the string immediately to the right. Similarly, when the model's belief that the current trial is either a word trial or a nonword trial crosses the decision threshold, the model will program the appropriate motor response. Figure 1 shows how this process works schematically.

The strategies that the model may employ are completely determined by the values of the saccade and decision thresholds. Higher thresholds require the model to have more confident beliefs before deciding to saccade or make a motor response, and thus will produce more accurate responses. However, lower thresholds require the model to collect fewer samples before being able to make a decision, and thus will produce faster reaction times. Therefore there is a speed-accuracy tradeoff, and not all values of these thresholds will perform equally well in the three payoff conditions. Lewis et al. used Monte Carlo simulations to find the values of these thresholds that produced the highest expected payoffs in each condition (see the method section for details), and they found that the single fixation durations (SFDs—fixation times of words that are only read once) produced by the model were similar to those of the humans they tested.

The New Policy Space

Although the original formulation of this model was successful in producing fixation durations similar to those of humans, it cannot explain regressions because saccades are always made to the string immediately to the right, so a new mechanism must be implemented for choosing which string to saccade to. There are many possible ways to do this. The Bounded Optimal model could adapt the various states of lexical access from E-Z Reader, or SWIFT's distinct mechanisms for choosing where and when to saccade, but in the interests of parsimony and computational tractability, a mechanism that requires the fewest parameters and requires minimal changes from the original model should be chosen.

The policy space of the new model is identical to the old except that saccadic control is no longer governed by thresholds. Instead the model follows a simple rule similar to that of Mr. Chips: the model always targets the string with the most uncertainty, i.e., the string whose belief is closest to 0.5. At each time step, if the currently fixated string's belief is closest to 0.5, the model will remain fixated there. Otherwise, it will program a saccade to the string whose belief is closest to 0.5. Once a saccade has been programmed, it cannot be cancelled or target to a new string, even if the targeted string's belief is no longer closest to 0.5 after saccade programming has completed. If two strings' beliefs are equally close to 0.5, the one closest to the current fixation will be chosen.

In order to discourage the model from frequently making very long saccades, however, a new parameter is introduced: the distance weight. The distance of each string's belief from 0.5 is adjusted according to the following function: *adjusted distance* = *abs*(*belief* – 0.5) + *distance weight* * *distance from current fixation*, where the distance from the current fixation is measured in number of strings. Figure 2 illustrates this policy in action.

The distance weight functions similarly to the saccade threshold in that a higher saccade weight requires the model to collect more information about a string before it can move on to the next, so a similar speed-accuracy tradeoff exists. However the nature of this tradeoff is more nuanced with the distance weight, because although a lower saccade threshold may allow the model to saccade away from a string before it has collected sufficient information about it, that lower distance weight will also make it easier for the model to return to that string later. The ability to “double-check” a string may lead to greater accuracy, but may also waste valuable time, so like the saccade and decision thresholds, this parameter must be optimized individually for the speed, accuracy, and balanced conditions.

Now that the model has been outlined, I will detail the behavioral experiments used to extract data from human participants, how parameters in the model were optimized, and examine how well their results matched.

Method

Participants

107 members from the University of Michigan community participated in this experiment. 61 were paid a baseline of $10 for participation, plus a bonus of $1 for each 1000 points they earned in the task. The remaining participants earned course credit for their participation. Data from 29 were unusable due to failure to complete the experiment, eye-tracker calibration problems, or other equipment malfunctions, leaving a total of 78 participants.

Procedure

After completing the informed consent form, participants were sat in front of a CRT monitor and an Eyelink I head-mounted eye-tracker operating at 250Hz was placed on their heads. After the eye-tracker was calibrated, participants were given instructions for completing the LLDT. Each participant was assigned to one of the payoff conditions described previously, but they were not told the name of their condition. They were only given a description of their task requirements, e.g., ”You will receive a point for each 125 milliseconds by which your response is faster than 5000 ms (5 s). You will lose 150 points if your response is incorrect. You will get a $1 bonus for each 1000 points."

Participants responded to 200 trials of the LLDT divided into 10 blocks preceded by a 10-item practice block. Half of the trials contained all words, and the other half contained 5 words and one nonword. Word and nonword trials were dispersed randomly throughout the experiment. In order to correct for drift in the eye-tracker, participants had to fixate on a dot in the center of the screen before every trial. After drift correction, participants then had to fixate on a dot in the leftmost string position to make the list of strings appear. This ensured that all participants would start reading from the left. Responses were recorded using a Cedrus response box.

Words were taken from a 234-word subset of the Brown Corpus (Kucera & Francis, 1967) which contained 117 high-frequency words (mean frequency count 239.2, SD 186) and 117 low frequency words (mean frequency count 5.6, SD 12.8). There were 53 different nonwords in total, and all were pronounceable according to English phonotactics. Items were presented on a CRT monitor in a 20pt Courier font and were separated by 8 characters of whitespace. At 25 inches from the screen, each word covered 0.7 inches, or 1.6 degrees of visual angle, and whitespace covered 1.48 inches, or 3.4 degrees of visual angle.

Model Simulations

In order to find the optimal policy for the model in each condition, all combinations of values within specific ranges were tested for the distance weight, decision threshold, and standard deviation of the perceptual noise. Search ranges for each value can be found in Table 6. For each combination of values, 10,000 trials were simulated, and the expected value of each policy was computed by taking the mean payoff of those trials. The policies with the highest expected values for each noise level were then fit to the human data by choosing the noise level which minimized the root mean squared error from SFDs for the three payoff conditions. The model was then simulated again for 100,000 trials using only these derived parameters in order to get highly accurate estimates for its SFDs and regression behavior.

As in the human experiment, each trial in the simulation had a 50% chance of being either a word or a nonword trial. The strings in each trial were drawn from a list of 500 words and 500 nonwords. For every parameter combination, new lists were generated. This was done to ensure that the model's results were not due to any particular choice of lexicon.

Results

Trial Level Results in the Human Data

Data analysis for the human data was conducted using mixed effects logistic regressions (Pinheiro & Bates, 2000), using the lme4 package for the R environment for statistical computing (R Development Core Team, 2011). Models with maximal random effects structures were fit. Trials with reaction times of 250ms or less were excluded from the analysis, but no other data were excluded. Hypothesis tests involving payoff condition were conducted using a single pair of normalized orthogonal contrasts. The only contrast of interest is the contrast between the speed and accuracy payoffs (i.e., accuracy and speed were coded as ±0.5, and balanced as 0); the second contrast is included for orthogonality but is not theoretically informative, so it is not reported. Under this design the balanced condition exists to improve error estimates and improve statistical power. All results are reported at the .05 significance level.

In the trial level analysis, the dependent variable was the probability that the trial would contain at least one regression. The model was fit with the following variables used as random intercepts: participant, trial number, block, and whether or not the participant was paid. The following variables were used as random slopes: payoff condition, trial type (word trial or nonword trial), trial correctness, and the interactions between all these terms.

Complete trial level results can be found in Table 3 with a graphical illustration in Figure 3a. Significant main effects were found for trial type and trial correctness, with regressions being more likely in word trials and in incorrect trials. Significant interactions were found between trial type and payoff condition and trial type and trial correctness, with regressions being more likely in word trials in the accuracy condition and in incorrect word trials generally. A significant three-way interaction was found between payoff condition, trial type, and trial correctness; a crossover interaction between trial type and correctness exists in the accuracy condition but not the speed condition.

String Level Results in the Human Data

Two separate analyses were performed at the individual string level. Both the probability of making a regression from a string, and the probability of making a regression to a string were used as dependant variables. Participant, trial number, and block were used as random intercepts. Payoff condition, string type (word or nonword), word frequency (coded as high frequency or low frequency), trial type, trial correctness, string position, the interaction between payoff condition and string frequency, and the interaction between payoff condition and string position were all used as random slopes. Each string position was coded as a separate categorical value, labeled 1-6, with 1 being the leftmost string and 6 the rightmost. For the analysis on regressions from strings, hypothesis tests about string position were conducted by individually contrasting positions 3, 4, 5, and 6 with position 2. For the analysis on regressions to strings, positions 2, 3, 4, and 5 were each individually contrasted with position 1. Positions 2 and 1 served as baselines because they are the first positions that can be regressed from and regressed to, respectively (note that a regression cannot be made from position 1, and a regression cannot be made to position 6).

For the analysis on regressions from strings, complete results can be found in Table 4, and a graphical representation can be found in Figure 4a. As in the trial level analysis, significant main effects were found for trial type and trial correctness, with regressions being more likely in word trials and in incorrect trials. Main effects were also found for string type, with regressions from nonwords being more unlikely than regression from words, and main effects were found for string position, with regressions from position 3 being more unlikely from position 2, but with regressions from positions 4, 5, and 6 being more likely. A significant interaction was found between payoff condition and string position, with regressions being more likely to occur from position 6 in the accuracy condition.

For the analysis on regressions to strings, complete results can be found in Table 5, with a graphical representation in Figure 5a. Again significant main effects were found for trial type and trial correctness, with regressions being more likely in word trials and in incorrect trials. Main effects were found for string type and word frequency, with regressions being more likely to be made to nonwords and low frequency words. A main effect was for string position, with regressions to string positions 2, 3, 4, and 5 being more likely than regressions to position 1. Unlike in the other analyses, a main effect for payoff condition was also found, with regressions being more likely in the accuracy condition. Significant interactions were found between payoff condition and string position, with regressions to string positions 3, 4, and 5 being less likely in the accuracy condition.

Model Results

The optimal distance weight and decision threshold for each condition are reported in Table 6, as well as the noise level which best fit the human results. A comparison of between the SFDs in the human data and the model data can be found in Table 7. Results from the model's simulations are displayed graphically in Figures 3b, 4b, and 5b. Statistical tests are not reported for these results because with 100,000 simulated trials, the confidence intervals around the means are negligible.

Discussion

The results from the human data confirm that task demands do indeed affect people's regression behavior. Figure 3a indicates that humans are using similar strategies in the accuracy and balanced conditions. They are more likely to make regressions in correct nonword trials than in incorrect nonword trials, and they are more likely to make regressions in incorrect word trials than in correct word trials. This means that people are more likely to make regressions when they think they saw a nonword. This strategy makes sense. If a person gets to the end of a trial and is still unsure about whether it was a word or a nonword trial, the most sensible thing to do is look back at any strings that might have been nonwords.

A different strategy seems to be going on in the speed condition. People are somewhat more likely to make regressions in incorrect trials than in correct trials in this condition, but this effect is consistent between the two trial types. The "double-checking" behavior in the accuracy and balanced conditions would probably waste too much time in the speed condition to be effective. Regressions in the speed condition probably happen when people rush through the trial too quickly and need to look back at some words to gain enough confidence to make a decision.

More regressions happen in incorrect trials because incorrect trials reflect a greater level of uncertainty.

One thing that is consistent among all conditions is that more regressions occur in word trials than in nonword trials. This is counterintuitive since Figure 5a indicates that regressions are most likely to be made to nonwords, but there might actually be more suspect nonwords in word trials. If a person encounters a string like “pago,” which does not closely resemble any common real words, he or she will likely conclude that it is a nonword quickly and make a response. However if a person encounters a very low frequency word like “helm” or “hilt,” he or she might be uncertain about whether it is a word or a nonword, because although such words are occasionally found in writing, they are rarely used in everyday speech. A reasonable strategy in this situation might be to look through the rest of the list to see if any other strings are obviously nonwords, and if no such words are found, go back to the suspect string to see if it was misread the first time around. This interpretation is consistent with the fact that 7 out of the 10 most regressed to strings are in fact very low frequency words, as can be seen in Table 8.

The model predicts a very different pattern of data. Unlike the humans the model is less likely to make a regression when it thinks it has seen a nonword. This may be due to the fact that unlike the humans, the model knows exactly which strings are words and which are nonwords. So when the model encounters a nonword and gets good information about it, it will quickly decide that it is in a nonword trial without wasting any time looking at the other strings in the trial. But since humans are less certain about what strings are and are not words, even when they get good information about a string and realize that it is not a word in their lexicon, they still may prefer to see what the other strings are before making a decision about the trial. This may be a good strategy because if humans are unsure about the identity of a string early in the trial but encounter a string later in the trial that they know for certain is a nonword, they will be able to infer that the earlier string actually was a word, and that information will allow them to perform better in subsequent trials.

The fact that the model know for certain which strings are and are not words also means that all the regressions produced by the model are not due to the "double checking" behavior seen in the humans but rather due to rushing through the trial too quickly and needing to collect more information before a trial decision can be made. Since the optimal policy in each condition has a distance weight of 0, the model will almost always program a saccade away from a string as soon as it receives information that that string is a word, because that string will usually no longer be the string with the most uncertain belief. However, if the model receives information that a string is a nonword, this will usually increase the string's uncertainty, so the model will continue to fixate on that string. The net result is that the model will rush through word trials very quickly and need to make lots of regressions, but will take its time on nonword trials and respond without looking at the other strings and without making any regressions. Having a distance weight of 0 in every policy leads to another limitation in the model: the differences in SFDs seen in the human data across the payoff conditions are virtually nonexistent in the model, as can be seen in Table 7. Furthermore the SFDs predicted by the model are not very accurate, being 55 ms off of the human SFDs on average.

Two important qualities of the human data are captured by the model, however. The first is that the model correctly predicts that regressions will be most likely to occur on the final string. This finding, however, says more about the LLDT than natural reading, however. What is more interesting is how the position effect adapts to each payoff. The humans are more likely to regress in the accuracy condition in the speed condition, but this is only true for correct word trials in the model. The model is *less* likely to make regressions in the accuracy condition for incorrect nonword trials.

The second quality that was captured was that regressions will be more likely to be made to nonwords than words (although this effect is not nearly as pronounced as in the human data). This finding is of relevance to natural reading. If a regression is made to a nonword, that indicates that the nonword was misidentified the first time it was fixated on (because otherwise the model would have responded with “nonword trial” right away). This indicates that regressions are induced by difficulties in linguistic processing, which is consistent with the predictions of E-Z Reader and SWIFT.

One change to the model that might bring its behavior more in line with the humans would be to decouple the decision of when to make a saccade from where to make a saccade, as in SWIFT. This change might reproduce the payoff effects on SFDs seen in the original model, and this may make the model more physiologically plausible as well, because there is some evidence that the decisions of where and when to saccade are mediated by different neural pathways (Carpenter, 2000; Findlay & Walker, 1999; Wurtz, 1996).

In order to test these hypotheses, a new version of the model was run using methods identical to those described above. This implementation of the model has both saccade thresholds and trial decision thresholds as in its original formulation, but instead of automatically saccading to the next string after crossing the saccade threshold, the model now chooses the string with the most uncertainty, as in the prior formulation, and it may continue to make saccades even after reaching the final string in the trial. The derived policy parameters of this model can be found in Table 9, a comparison of its SFDs to the human data can be found in Table 10, and graphical summaries of its regression behavior can be found in Figure 6.

The results indicate that this model is actually a worse fit to the human data in several aspects. Although there is a noticeable effect of payoff on SFDs this time, this model's SFDs are still overall a poor fit to the human data, with an average error of 50 ms. And unlike the last model, this model is no more likely to make regressions to nonwords than words, and the position effects seen in this model are not as strong as those in the last, and the interaction between position and payoff is opposite of what is seen in the human data. This version of the model still does not produce any "double-checking" behavior. It only makes regressions when it moves through the trial too quickly, and since it spends more time on each string in the accuracy and balanced conditions, it is more likely to make regressions in the speed condition.

This second model does have one advantage over the last model, however. Because this model is a generalization of the original model that allows for regressions, it can be used to test Bicknell and Levy's claim that regressive policies are faster and more accurate than nonregressive policies. Mean trial reaction times, percent of trials correct, and mean trial payoff can be found in Table 11 for both this model (the regressive model) and the model originally presented in Lewis et al. (the non-regressive model). While the regressive model is more accurate than the non-regressive model in all three conditions, it is only faster in the accuracy condition, and thus its payoff is also only higher in the accuracy condition. This means that the policy space of the regressive model must not be a superset of the policy space of the non-regressive model, because otherwise the regressive model would perform at least as well as the non-regressive. Therefore, given a choice between the target selection mechanism of the new models and the forced forward saccade mechanism of the original model, a rational agent should choose the latter. This does not imply that the humans in this study were irrational for making regressions, however, since the model's oculomotor architecture clearly does not have the same bounds as the humans'. But this finding does imply that Bicknell and Levy's claim that regressive strategies are better than non-regressive strategy depends on the specific type of model being tested and is not true in general.

There are two changes to the model that could make its behavior more similar to the humans'. The first change would be for the model to no longer have a clear distinction between words and nonwords, but instead to have a probabilistic belief about each string being a word. As discussed earlier, humans probably condition their belief that the current string is a nonword on their belief that the other strings are nonwords, so this change might cause the model produce some of the "double checking" behavior seen in the human data.

The second change would be to make this model a "leaky, competing accumulator" model (Usher & McClelland, 2001). At present, the model has perfect memory for its beliefs about each string in the current trial. This is probably unrealistic, though, because the human participants have to complete hundreds of these trials, and memories for all those trials are likely to interfere with each other. When humans make a regression, it is probably because they cannot remember if they really saw a nonword in that trial or if they are thinking of another trial. This could be simulated in the model by having its beliefs about each string slowly degrade back to their priors. This would mean that once the model reaches the end of a trial, its belief about strings from the beginning of the trial will be more uncertain, so it will need to make regressions in order to cross the decision threshold.

Summary of Results and Conclusion

Being able to explain regressions is important for computational models of reading because it addresses the more fundamental question of how readers choose which word to read next and whether or not human reading strategies are rational. Many models have been put forward that try to explain regressions, but all have their own limitations. Both E-Z Reader and SWIFT propose that regressions arise from difficulties in linguistic processing, but E-Z Reader unparsimoniously requires separate mechanisms for regressions and forward saccades, and the mechanism for regressions in SWIFT is irrational. Bicknell and Levy's model proposes that regressions allow for faster and more accurate reading in a stochastic system, but it does not account for forward saccades. Mr. Chips proposes that regressions are an emergent phenomenon arising from a simple rule that tries to minimize uncertainty, but says nothing about fixation durations.

None of these models can account for possible effects of task demands on regression behavior. The Bounded Optimal Model was previously successful in explaining the effects of task demands on single fixation durations, and so it was modified to allow for regressions. The core of this model is bounded optimality, the notion that human behaviors are approximately optimal adaptations to the joint constraints of the human information processing system, the external probabilistic environment, and an internal reward function. This model examines the interaction between the high level cognitive processes of word recognition and decision making and the low level oculomotor processes of eye movements.

Both humans and computer simulations of the Bounded Optimal Model completed trials of the list lexical decision task, in which the goal was to decide if a list of strings were all real English words, or at most one of them was a nonword. Results from the human experiment presented in this paper showed that tasks demands, and in particular the tradeoff between speed and accuracy, affect peoples' regression behavior. The Bounded Optimal Model tried, and ultimately failed, to explain the humans' behavior, and it failed to corroborate Bicknell and Levy's claim that regressions allow for faster and more accurate reading. It seems that humans used two types of regressions in the LLDT. The first type were used to “double check” their work and to ensure that they did indeed see a nonword earlier in the trial; the second were used to compensate for rushing through trials too quickly and not acquiring enough information to make a decision. The model was only able to make the second type of regressions. Two possible changes to the model may cause it to produce the first type, however.. The first is to give the model more continuous, probabilistic beliefs about which strings are words and which are nonwords. The second is to introduce “leaks” into the model’s memory, so that its belief about each string will become less confident over time.

Works Cited

Anderson, J. R. (1990). *The Adaptive Character of Thought*. Hillsdale, NJ: Lawrence Erlbaum.

Bicknell, K., & Levy, R. (2010a). A rational model of eye movement control in reading. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 1168-1178.

Bicknell, K., & Levy, R. (2010b). Rational eye movements in reading combining uncertainty about previous words with contextual probability. *32nd Annual Conference of the Cognitive Science Society.* Austin, TX: Cognitive Science Society.

Blanchard, H. E., & Iran-Nejad, A. (1987). Comprehension processes and eye movement patterns in the reading of surprise-ending stories. *Discourse Processes, 10*, 127-138.

Carpenter, R. H. (2000). The neural control of looking. *Current Biology*, *10*, R291-R293.

Engbert, R., Longtin, A., & Kliegle, R. (2002). A dynamic model of saccade generation in reading based on spatially distributed lexical processing. *Vision Research, 42(5)*, 621-636.

Engbert, R., Nuthmann, A., Richter, E., & Kliegl, R. (2005). SWIFT: A dynamical model of saccade generation during reading. *Psychological Review*, *112(4)*, 777-813.

Findlay, J. M., & Walker, R. (1999). A model of saccade generation based on parallel processing and competitive inhibition. . *Behavioral and Brain Sciences*, *22*, 661-721.

Frazier, L., & Rayner, K. (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology, 14*, 178-210.

Hale, J. T. (2011). *What a rational parser would do. Cognitive Science, 35,* 399–443.

Harwood, M. R., Mezey, L. E., & Harris, C. M. (1999). The spectral main sequence of human saccades. *The Journal of Neuroscience, 19(2)*, 9098-9106.

Jacobson, J. Z., & Dodwell, P. C. (1979). Saccadic eye movements during reading. *Brain and Language, 8*, 303-314.

Kucera, H., & Francis, S. (1967). *Computational analysis of present-day American English.* Providence, RI: Brown University Press.

Legge, G. E., Hooven, T. A., Klitz, T. S., Mansfield, J. S., & Tjan, B. S. (2002). Mr. Chips 2002: new insights from an ideal-observer model of reading. *Vision Research, 42(18)*, 2219-2234.

Legge, G. E., Klitz, T. S., & Tjan, B. S. (1997). Mr. Chips: An ideal-observer model of reading. *Psychological Review, 104(3)*, 524-553.

Levy, R., Bicknell, K., Slattery, T., & Rayner, K. (2009). Levy, R., Bicknell, K., Slattery, T., & Rayner, K. (2009). Eye movement evidence that readers maintain and act on uncertainty about past linguistic input. *Proceedings of the National Academy of Sciences, 106(50)*, 21086-21090.

Lewis, R. L., Shvartsman, M., & Singh, S. (to appear). The adaptive nature of eye-movements in linguistic tasks: How payoff and architecture shape speed-accuracy tradeoffs. *Topics in Cogntive Science*.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review, 104*, 3-65.

Meyer, D. E., & Schvaneveldt, R. W. (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology, 90*, 22-34.

Norris, D. (2006). The Bayesian reader: Explaining word recognition as an optimal bayesian decision process. *Psychological Review, 113(2)*, 327-357.

Pinheiro, J. C., & Bates, D. M. (2000). *Mixed Effects Models in S and S-Plus.* Springer.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin, 123(3)*, 372-422.

Rayner, K., & Pollatsek, A. (1989). *The psychology of reading.* Englewood Cliifs, NJ: Prentice Hall.

Reichle, E. D., Pollatsek, A., & Rayner, K. (2006). E-Z Reader: A cognitive-control, serial-attention model of eye-movement behavior druing reading. *Cognitive Systems Research, 7*, 4-22.

Reichle, E. D., Warren, T., & McConnell, K. (2009). Using EZ Reader to model the effects of higher level language processing on eye movements during reading. *Psychonomic Bulletin & Review 16.1*, 1-21.

Reichle, E., Pollatsek, A., Fisher, D., & Rayner, K. (1998). Toward a model of eye movement control in reading. *Psychological Review, 105(1)*, 125-157.

R Development Core Team (2011). *R: A language and environment for statistical computing.* [Computer software manual]. Vienna, Austria. Available from http://www.R-project.org/: (ISBN 3-900051-07-0).

Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: the leaky, competing accumulator model. Psychological review, 108(3), 550.

VanRullen, R., & Thorpe, S. J. (2001). The time course of visual processing: From early perception to decsion-making. *Journal of Cognitive Neuroscience, 13(4)*, 454-461.

Wurtz, R. H. (1996). Vision for the control of movements. *Investigative Opthalmology & Visual Science, 37*, 2131-2145.

Author Note

Craig A. Sanders, Department of Psychology, University of Michigan, Ann Arbor

I would like to thank Dr. Richard Lewis for collaborating with me on this project, as this thesis would not have been possible if it were not for his guidance and support. I also give my thanks to Dr. Satinder Singh for lending his expertise, and I would like to give a special thanks to Michael Shvartsman for all his invaluable help with statistics, programming, debugging, and interpreting results. Finally I would like to thank Yasaman Kazerooni, Mehgha Shyam, Emmanuel Kumar, Bryan Berend, and all the other research assistants in the Language and Cognitive Architecture Lab that helped collect the data I used in this study.

Table 1

Quantitative description of each payoff condition

	Accuracy	Balanced	Speed
Incorrect penalty	-150	-50	-25
Speed bonus (per second under 5s)	8	6.7	5.7

Table 2

Model parameters

Parameter	Mean	Std Deviation	Source
Eye-brain lag	50	15	(VanRullen & Thorpe, 2001)
Saccade programming time	125	37.5	E-Z Reader (Reichle et al., 2009)
Saccade execution time	2.8 * (distance of saccade in degrees) + 24.7	0.3 * mean	(Harwood, Mezey, & Harris, 1999)
Motor preparation and execution time	100	30	EPIC (Meyer & Kieras, 1997)
Trial onset detection and refixation	150	45	Prior estimate of short fixation and saccade
Sample duration	10	0	Nontheoretical discretization parameter
Gaussian sample noise	0	1.5	Standard deviation fit as described in text

Note. Means were taken from sources noted. All units are in milliseconds.

Table 3

Trial level analysis of regressions

Predictor	Estimate	SE	*z*	*p*
Condition	0.16	0.25	0.63	.532
Trial Type	0.76	0.11	7.10	<.001
Correctness	0.22	0.06	3.51	<.001
Condition x Trial Type	0.64	0.27	2.38	.017
Condition x Correctness	0.23	0.16	1.44	.151
Trial Type x Correctness	-0.63	0.11	-5.57	<.001
Condition x Trial Type x Correctness	-0.63	0.28	-2.22	.026

Note. Coefficient estimates, standard errors, Wald's *z*-scores, and *p* values calculated using a mixed effects logistic model.

Table 4

Analysis of regressions from string.

Predictor	Coef. β	SE(β)	*z*	*P*
Condition	0.24	0.16	1.53	.123
Word Frequency	0.03	0.02	1.16	.246
String Type	-2.28	0.08	-27.23	<.001
Correctness	0.22	0.03	6.45	<.001
Trial Type	0.75	0.03	29.58	<.001
String Position 3	-0.1	0.04	-2.50	.012
String Position 4	0.10	0.04	2.43	.015
String Position 5	0.54	0.04	14.36	<.001
String Position 6	2.57	0.03	75.67	<.001
Condition x Word Frequency	-0.06	0.06	-1.10	.272
Condition x String Type	0.16	0.19	0.82	.411
Condition x String Position 3	-0.111	0.10	-1.18	.238
Condition x String Position 4	-0.11	0.10	-1.12	.261
Condition x String Position 5	-0.03	0.10	-0.40	.688
Condition x String Position 6	0.25	0.08	3.12	.002

Note. Coefficient estimates, standard errors, Wald's *z*-scores, and *p* values calculated using a mixed effects logistic model.

Table 5

Analysis of regressions to strings

Predictor	Coef. β	SE(β)	*z*	*P*
Condition	0.45	0.13	3.52	<.001
Word Frequency	-0.19	0.02	-8.38	<.001
String Type	2.96	0.06	48.61	<.001
Correctness	-0.18	0.03	-5.91	<.001
Trial Type	-0.57	0.03	-21.91	<.001
String Position 2	0.66	0.03	20.25	<.001
String Position 3	0.96	0.03	30.83	<.001
String Position 4	0.85	0.03	26.21	<.001
String Position 5	1.04	0.03	32.82	<.001
Condition x Word Frequency	0.09	0.06	1.68	0.093
Condition x String Type	0.01	0.12	0.10	0.918
Condition x String Position 2	-0.13	0.08	-1.60	0.109
Condition x String Position 3	-0.27	0.08	-3.58	<.001
Condition x String Position 4	-0.31	0.08	-3.88	<.001
Condition x String Position 5	-0.27	0.08	-3.54	<.001

Note. Coefficient estimates, standard errors, Wald's *z*-scores, and *p* values calculated using a mixed effects logistic model.

Table 6

Derived policy parameters.

	Search range	Derived Values		
		Accuracy	Balanced	Speed
Distance Weight	0 – 0.09 in 0.01 increments	0	0	0
Decision Threshold	0.709 – 0.999 in 0.01 increments	0.999	0.989	0.989
Standard Deviation of Perceptual Noise	1 – 3 In 0.25 increments	1.5		

Table 7

Single fixation durations in the human and model data

		Human	Model	RMSE
Accuracy	Nonword	383	400	30
	Low Frequency Word	274	299	53
	High Frequency Word	256	274	36
Balanced	Nonword	324	404	84
	Low Frequency Word	256	319	80
	High Frequency Word	242	272	43
Speed	Nonword	358	404	53
	Low Frequency Word	268	319	78
	High Frequency Word	253	272	36

Note. Means and root mean squared errors are reported in milliseconds.

Table 8

The 10 most regressed to strings in the human data

String	Number of Regressions	Frequency
garb	105	3
pith	102	1
bard	101	3
prim	84	1
bout	82	7
stad	79	0
nigh	79	1
meck	78	0
fore	76	7
fets	74	0

Note. Frequency is reported as number of occurrences out of 1,000,000 words (Kucera & Francis, 1967). 0 indicates a nonword.

Table 9

Derived policy parameters for the second model

	Search range	Derived Values		
		Accuracy	Balanced	Speed
Distance Weight	0 – 0.09 in 0.01 increments	0	0	0
Saccade Threshold		0.919	0.919	0.879
Decision Threshold	0.709 – 0.999 in 0.01 increments	0.999	0.999	0.999
Standard Deviation of Perceptual Noise	1 – 2 In 0.25 increments		1	

Table 10

Single fixation durations in the human and the second model data

		Human	Model	RMSE
Accuracy	Nonword	383	416	46
	Low Frequency Word	274	290	43
	High Frequency Word	256	280	45
Balanced	Nonword	324	416	98
	Low Frequency Word	256	290	53
	High Frequency Word	242	280	54
Speed	Nonword	358	378	41
	Low Frequency Word	268	261	41
	High Frequency Word	253	242	29

Note. Means and root mean squared errors are reported in milliseconds.

Table 11

Mean reaction times,% correct, and mean payoff of regressive and non-regressive models

	Reaction Time (ms)		% Correct		Mean Payoff	
Condition	Non-regressive Model	Regressive Model	Non-regressive Model	Regressive Model	Non-regressive Model	Regressive Model
Accuracy	1644	1569	98	99.49	23.44	28.456
Balanced	1546	1569	97	99.49	21.52	19.893
Speed	1455	1637	95	99.37	18.89	15.957

Note. Data for the Regressive Model comes from the second model described in the discussion.

Data for the Non-regressive Model comes from Lewis et al. (to appear).

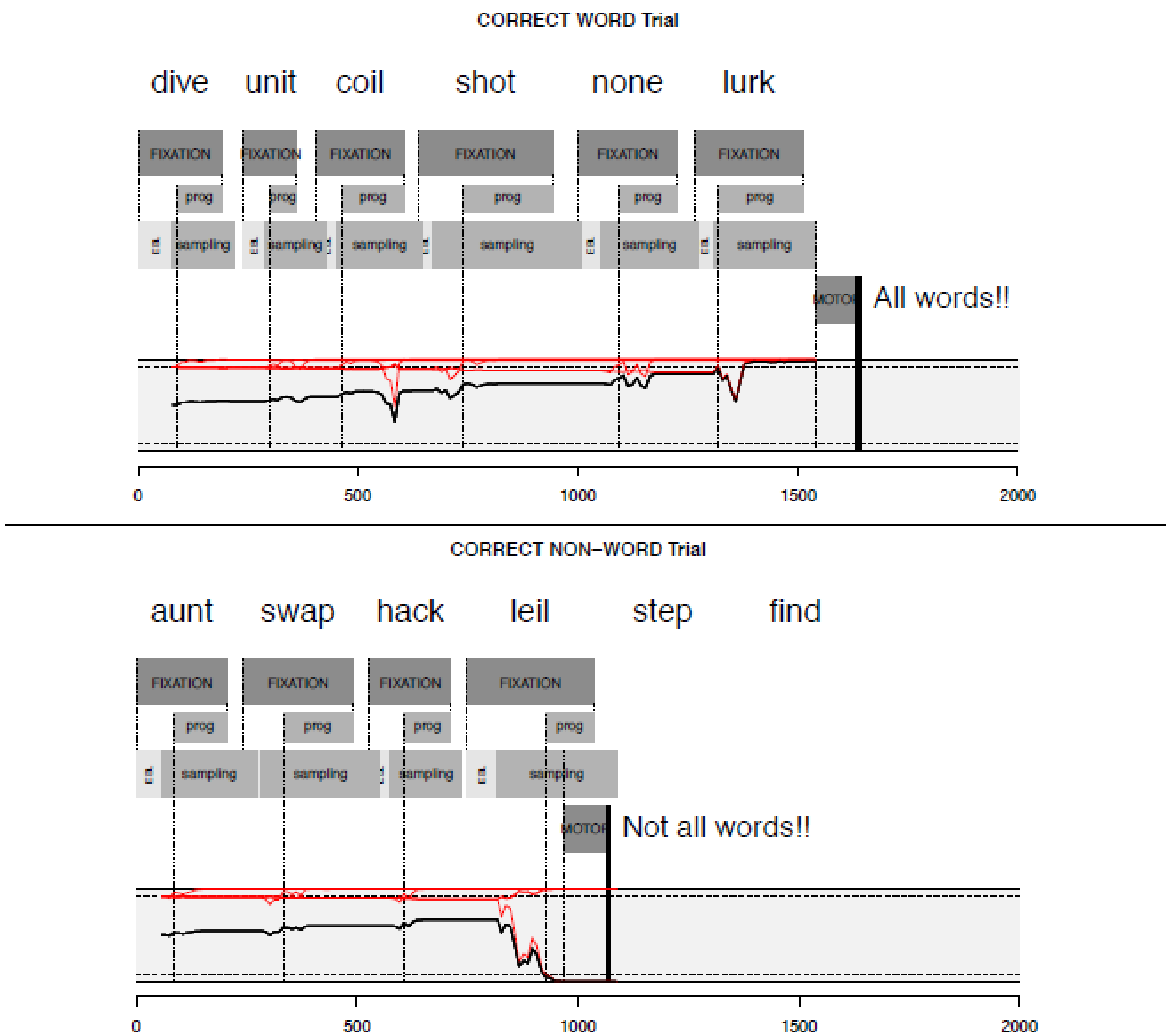


Figure 1. Simulated correct word trial and correct nonword trial. The strings at the top are the strings presented in the trial. The filled rectangles show the timing and duration of fixation durations, saccade programming (prog), eye-brain-lag (EBL), sampling, and motor response and execution. At the bottom is the random walk of the belief probabilities, with the bottom representing 0 and the top 1. The black line is the belief that the trial is a word trial (starting at 0.5), and the red lines are the beliefs that each string is a word (starting at 0.92). The solid black lines represent the decision thresholds, and the dashed lines represent the saccade thresholds. Adapted from (Lewis, et al., to appear).

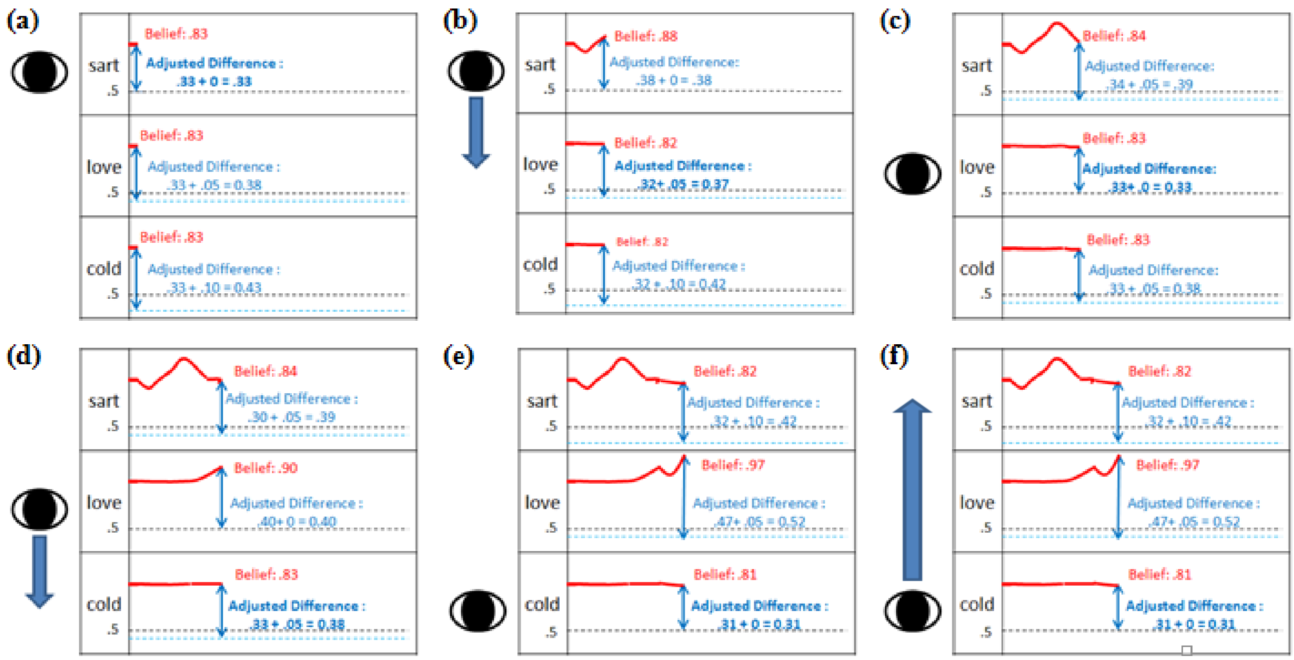
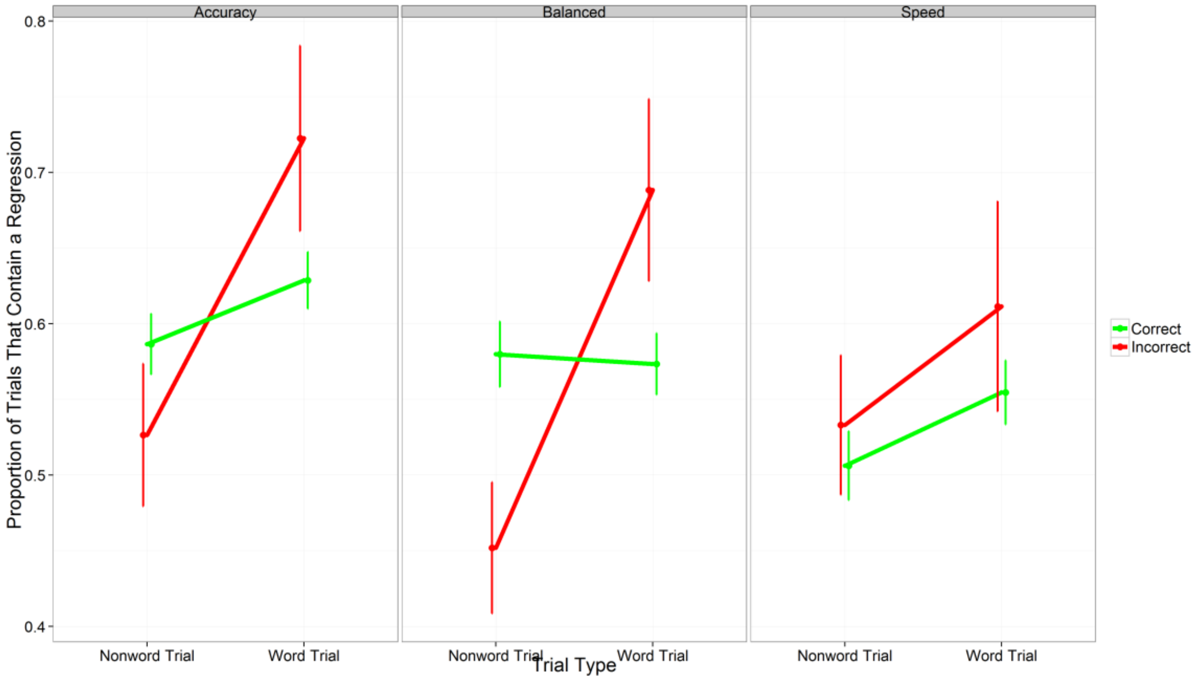


Figure 2. Simulated trial of 3 strings under new policy space with a distance weight of 0.05. (a) The model begins the trial fixated on the nonword "sart." The beliefs of all strings start out equal, but the belief of "sart" has lowest adjusted difference from .5, so the model remains fixated there. (b) After collecting some noisy information, the model incorrectly starts to believe "sart" is a word. The lowest adjusted difference is now at "love" so the model programs a saccade. (c) The model executes a saccade to "love." During saccade programming, the model's belief about "sart" decreased. (d) The model begins to believe that "love" is a word, and the lowest adjusted difference is now at "cold," so a saccade is programmed. (e) The model executes a saccade to "cold." During saccade programming, the model became even more confident that "love" is a word. (f) After receiving some noisy information, the model begins to believe that "cold" is a word. "Sart" now has the lowest adjusted difference, so a regression is programmed back to it.

(a)



(b)

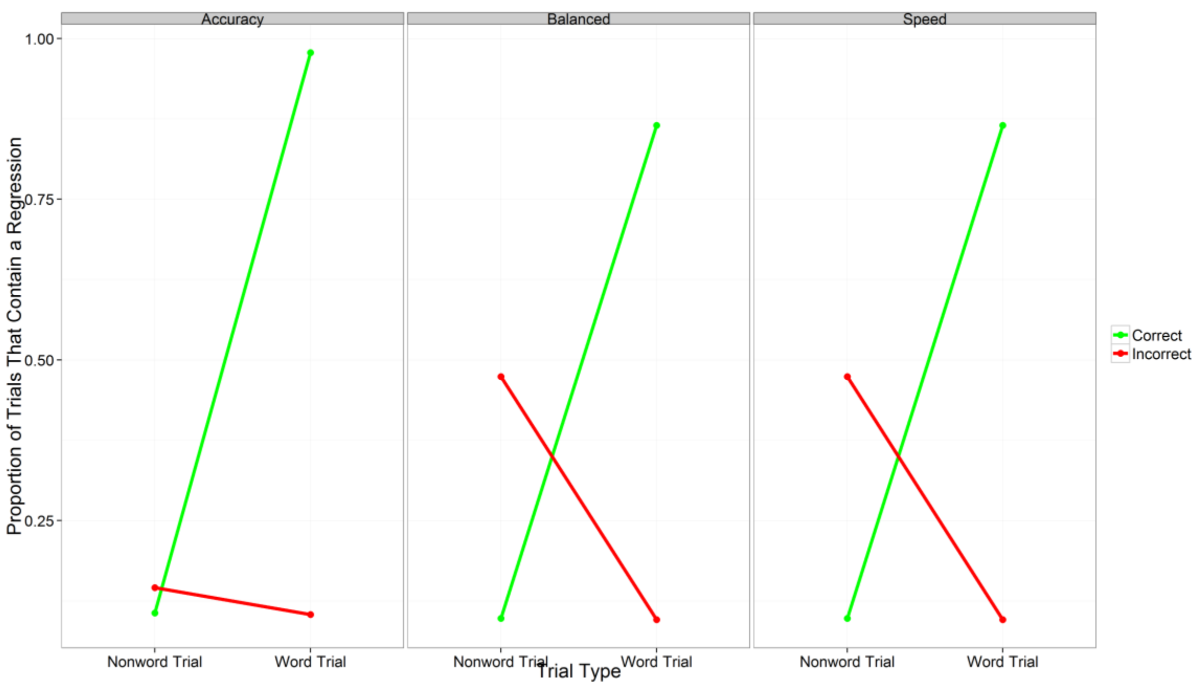


Figure 3. Proportion of trials that contain at least one regression, divided by payoff condition, trial type, and correctness. Human results are shown in (a) and model results are shown in (b).

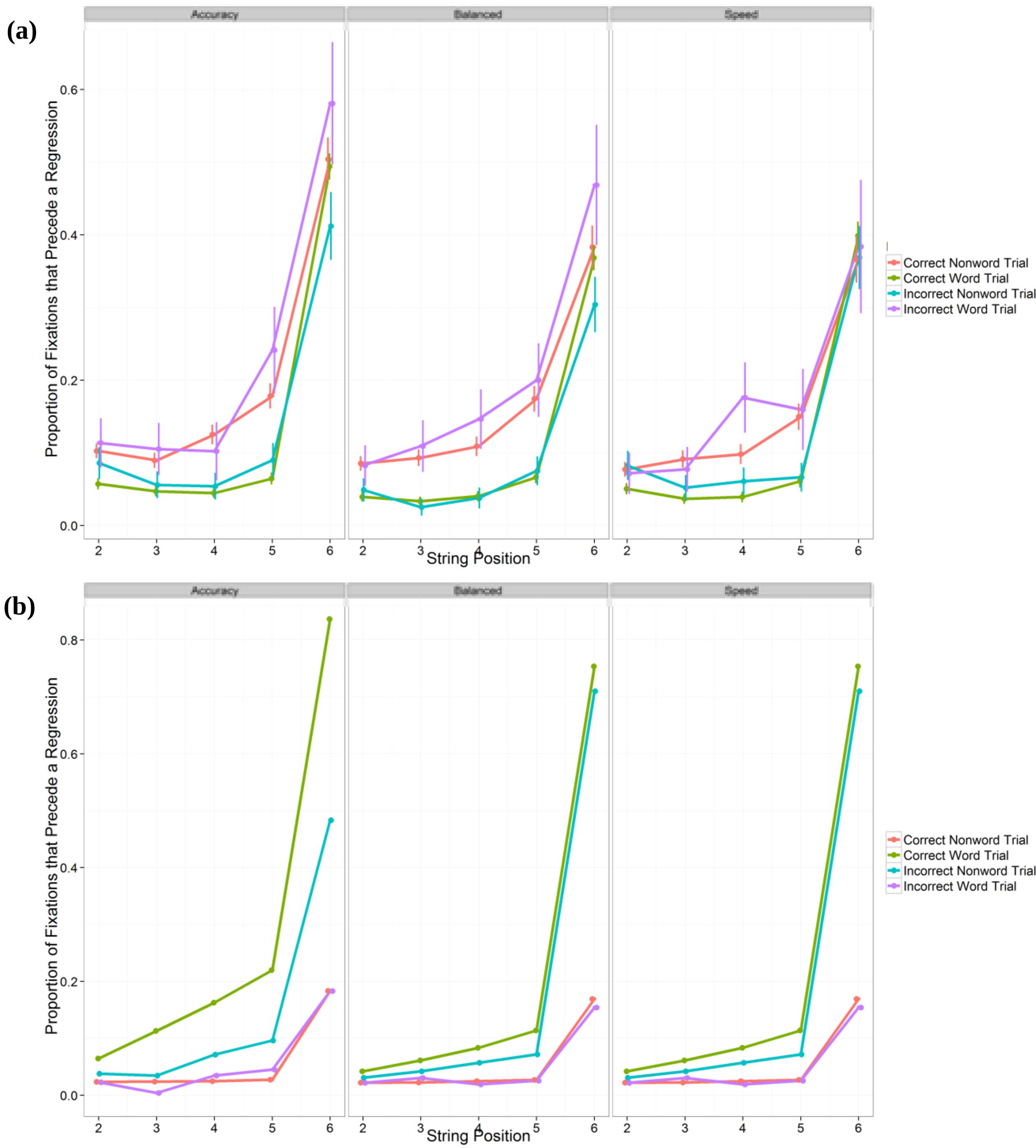


Figure 4. Proportion of fixations on each string position that precede a regression. Strings are ordered 1-6, with 1 being the leftmost string and 6 the rightmost string. String 1 is not shown because a regression cannot be made from that position. Human results are shown in (a) and model results are shown in (b).

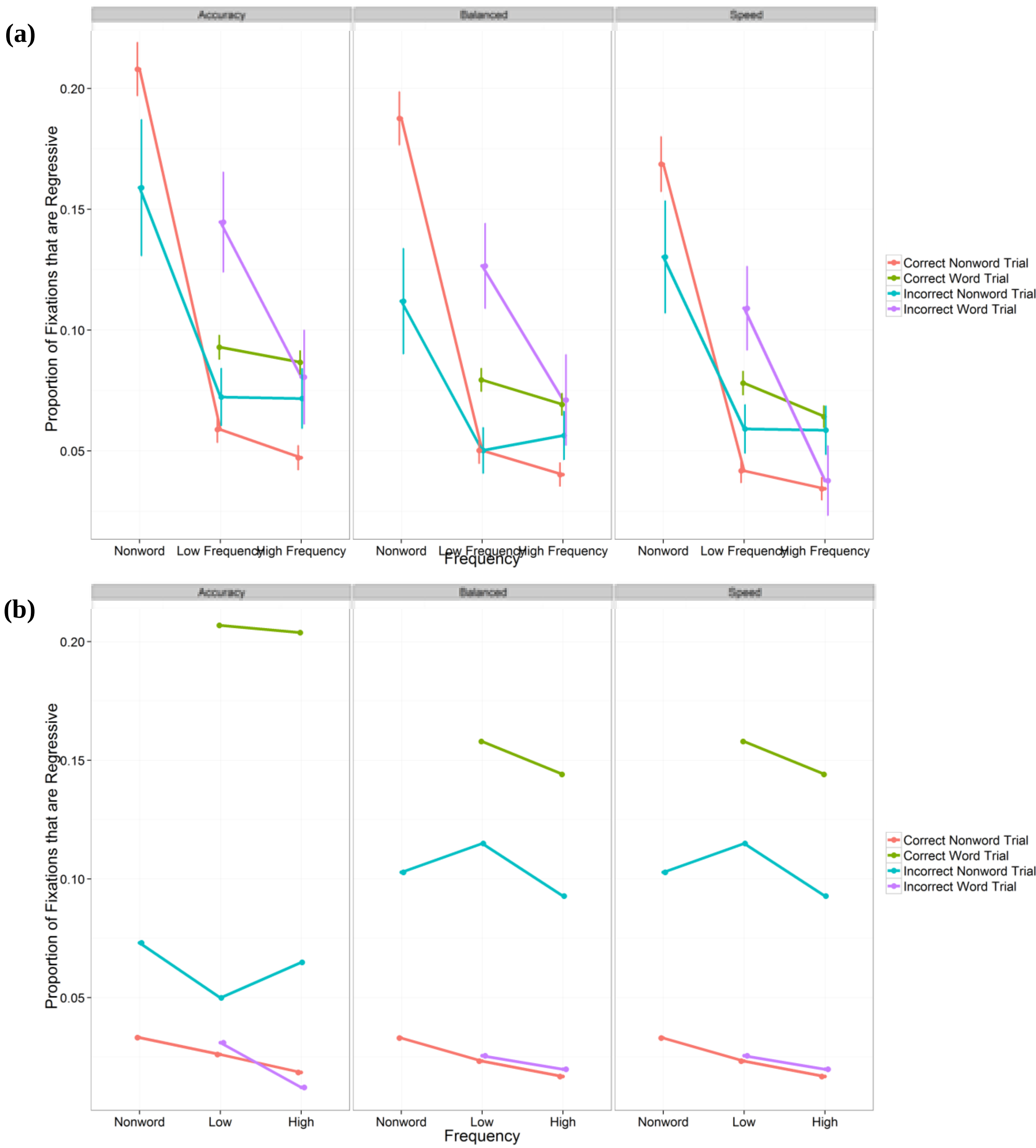


Figure 5. Proportion of fixations on string frequency level that are regressive. Human results are shown in (a) and model results are shown in (b).

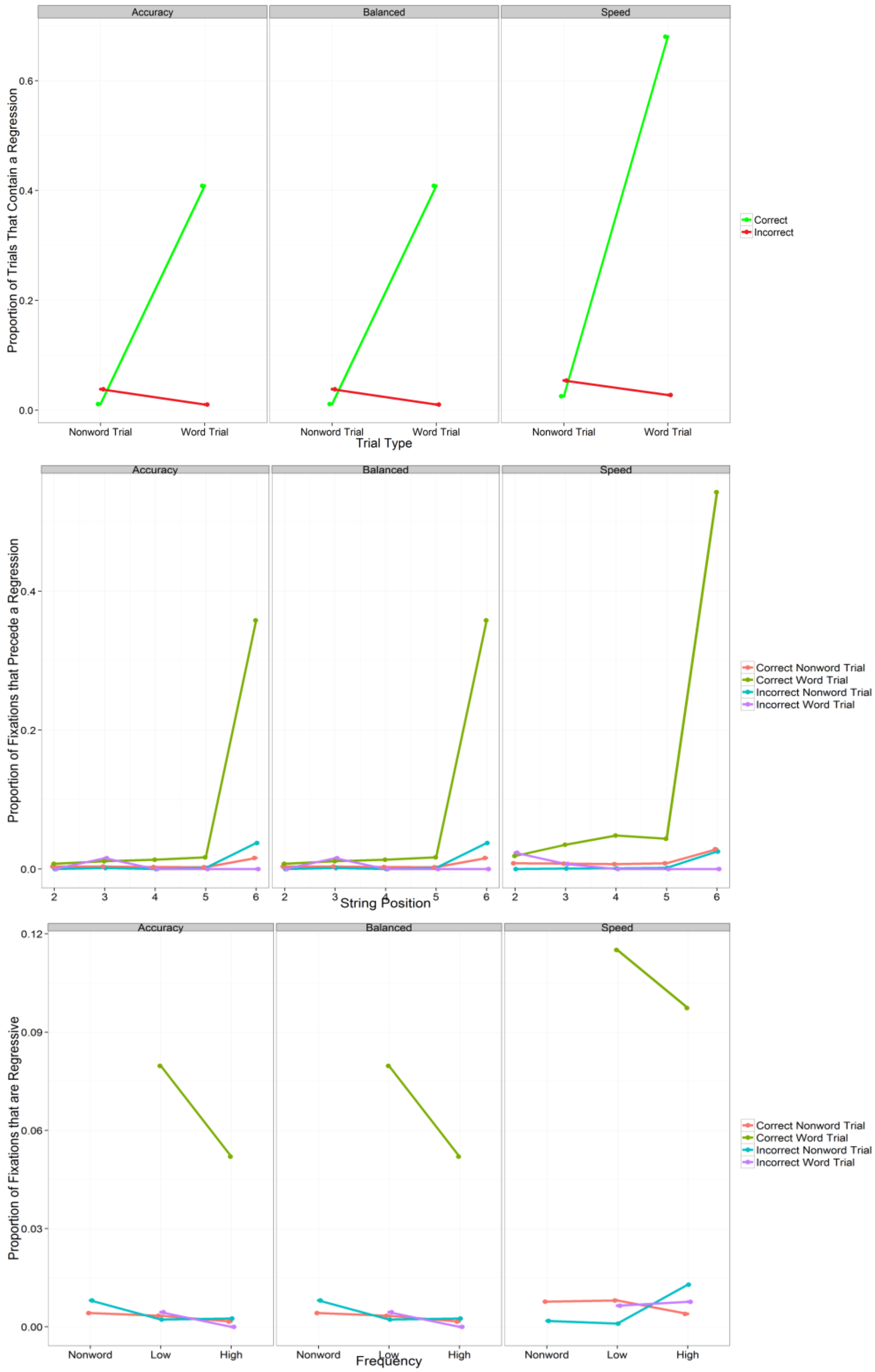


Figure 6. Regression behavior in the second model. (a) Overall frequency of regressions divided by trial type and correctness. (b) Proportion of fixations on each string position that precede a regression. (c) Proportion of fixations on each string frequency that are regressive.