Towards a Complete Model of Reading: Simulating Lexical Decision, Word Naming, and Sentence Reading with Über-Reader

Aaron Veldre (aaron.veldre@sydney.edu.au)
School of Psychology, The University of Sydney, NSW 2006, Australia

Lili Yu (lili.yu@mq.edu.au)
Department of Cognitive Science, Macquarie University, NSW 2109, Australia

Sally Andrews (sally.andrews@sydney.edu.au)
School of Psychology, The University of Sydney, NSW 2006, Australia

Erik D. Reichle (erik.reichle@mq.edu.au)
Department of Psychology, Macquarie University, NSW 2109, Australia

Abstract

This paper presents simulations of eye movements during reading, lexical decision, and naming using Über-Reader, a new computational model that aims to provide a complete account of the perceptual, cognitive, and motor processes involved in reading. The present simulations focused on Über-Reader's word-identification module—an implementation of the Multiple-Trace Memory model (Ans et al., 1998) based on the theoretical assumptions of the MINERVA 2 model of episodic memory (Hintzman, 1984)—with a vocabulary comprising the full corpus of the English Lexicon Project (Balota et al., 2007). The model's lexicon was probed with words and one-letter-different non-words from the Schilling et al. (1998) corpus, and outputs of the model were scored to evaluate performance against the empirical data. The outcomes of these simulations will inform further development of Über-Reader by providing the foundation for our ultimate goal of simulating reading, in its entirety.

Keywords: computer modeling; eye movements; reading; lexical-decision task; naming task; visual word recognition; Über-Reader

Introduction

Reading has been the focus of research since the inception of psychology (e.g., Huey, 1908) and has proved to be a fertile area for computational modeling (see Reichle, 2020). To date, the models that have been developed to explain the representations and processes involved in reading are typically limited to a single level of the reading system. There has been a particular focus on modeling isolated word recognition: over 40 such models have been published in *Psychological Review* since 1980. This is perhaps unsurprising because words are the building blocks of meaning and provide the bridge between spoken and written language. Computational models of higher-order reading processes have also been developed, but their scope is typically limited to the construction of sentence or discourse representations. Although these models have stimulated the growth of vast empirical literatures, further theoretical progress requires an integrative approach to modeling that addresses fundamental questions about the *architecture* of the reading system (i.e., how the various components are organized and coordinated during natural reading; Andrews & Reichle, 2019).

Computational models of eye-movement control currently provide the closest approximation to models of the reading architecture. Unfortunately, existing eye-movement models fail to provide a detailed account of any of the component processes of reading. For example, one of the most successful of these models, *E-Z Reader* (Reichle et al., 1998; Reichle et al., 2012), has been used to simulate all of the major benchmark findings related to the eye movements of both skilled and developing readers. However, its utility is intrinsically limited because it only describes how key variables (e.g., word frequency) affect the time required to complete various processes (e.g., word identification) without providing a deep (computational) account of these processes. Importantly, E-Z Reader shares this limitation with other major models of eye movements (e.g., *SWIFT*; Engbert et al., 2005) that provide comparable fits to empirical data. This makes it difficult to adjudicate between competing theoretical claims, such as whether lexical processing occurs serially—for one word at a time—or in parallel, for multiple words simultaneously.

The overarching goal of the current project is to develop and test a new computational model of reading in its entirety—*Über-Reader* (Reichle, 2020; see Figure 1). The full Über-Reader model embeds components that process sentences (Van Dyke & Lewis, 2003) and discourse (Kintsch, 1998) within the framework of the E-Z Reader model which coordinates the movements of covert and overt attention. The assumptions of the model are based on general principles of memory, attention, and language processing, reflecting the fact that the reading system "piggy backs" on brain structures that evolved for other functions. Thus, in principle, the model should be able to simulate all of the tasks used to study reading, including both on- and off-line behaviors (e.g., lexical decision, word naming, patterns of eye movements during reading, and the recall of propositional content), without making any reading-specific assumptions.

The first stage of this project—and the specific goal of this paper—is to evaluate Über-Reader's word-identification component. To do this we tested the model's performance in simulating the most commonly used tasks to study isolated word identification: lexical decision and naming. Before validating the model's assumptions concerning sentence and discourse processing, it was also critical to verify that directly simulating the process of word identification did not compromise the model's ability to account for benchmark eye-movement effects observed in sentence reading.

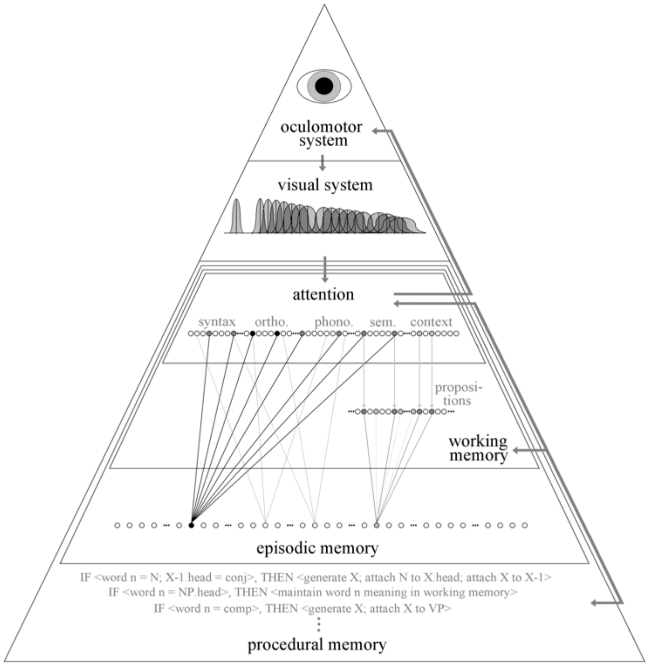
oculomotor system
visual system
attention
syntax
ortho.
phono.
sem.
context
proposi-tions
working memory
episodic memory
IF <word n = N; X-1.head = conj>, THEN <generate X; attach N to X.head; attach X to X-1>
IF <word n = NP.head>, THEN <maintain word n meaning in working memory>
IF <word n = comp>, THEN <generate X; attach X to VP>
procedural memory

Figure 1: Schematic diagram of the full Über-Reader model. During reading, episodic memory is probed with the orthographic features of the currently attended word to retrieve its pronunciation, meaning, and syntactic category. Sentences are parsed into phrase structures by a set of production rules in procedural memory. A composite pattern of the semantic features of key words from the sentence remains active in working memory and can pre-activate upcoming input. The pattern is encoded into long-term episodic memory when a sentence boundary is reached.

Overview of Über-Reader's Lexical Module

In Über-Reader, each new experience with a word is encoded as a discrete memory trace containing a collection of features representing the word's spelling, pronunciation, syntactic category, and contextually appropriate meaning. In the formalism of the model, these memory traces are vectors of elements that take on values of 1 or 0 to represent feature presence versus absence, respectively. For example, an encounter with the word "cat" would likely result in the encoding of a memory trace with the features corresponding to the letters "c," "a," and "t" in positions 1-3 being set equal to 1 and features corresponding to other letters being set equal to 0. Similarly, specific phonological features are encoded. Semantic features are defined by setting each of 500 possible features equal to 1 with a probability of p_{semantic}. Each word is thus represented by a small number (M = 10) of random features that denote aspects of the word's core meaning and likely case role(s).[1] Finally, syntactic features denote words belonging to seven possible parts of speech: adjectives, conjunctions, determiners, nouns, prepositions, verbs, and an "other" category. The model's lexicon comprises 40,411 words from the *English Lexicon Project* corpus (Balota et al., 2007).

Über-Reader retains the core assumption of E-Z Reader that attention is allocated to words in a strictly serial manner, supporting the lexical processing and identification of only one word at a time. The model's "engine" is the system that identifies printed words. A word can be identified by using its orthographic features to probe memory. This causes memory traces to "resonate" or become active to the degree that their orthographic features are similar to those in the probe. These active traces generate an overall familiarity signal, or *intensity*, and then settle into a stable pattern, called the *content*, that represents the words' spelling, pronunciation, meaning, and part of speech. This conceptualization of word identification is consistent with instance-based models, specifically the *Multiple-Trace Memory model* (Ans et al., 1998) that borrows extensively from *MINERVA 2*, a model of human memory (Hintzman, 1984).

Core Assumptions Relating to Word Identification

A memory trace containing the orthographic features of an encoded word probe will become active to the degree that the features of the trace are similar to those in the probe. Equation 1 describes how this similarity is calculated, where i is an index of the memory traces, j is an index of the N features, and N_r is the number of non-zero features in either the probe or trace. Because features are limited to taking on values of 1 or 0, the similarity between a probe and trace can range from 0 to 1, with the former indicating complete dissimilarity and the latter representing perfect similarity.

$$(1) \qquad similarity_i = \left(\sum_{i=1}^{N} probe_j trace_{i,j} \right) \Big/ N_r$$

The similarity values provided by Equation 1 can be used to gate the memory traces that will contribute to information that is ultimately retrieved from memory (see Dougherty et al., 1999). This gating is done using a minimal similarity threshold, $\theta_{\text{similarity}}$, to delimit those traces that will become active in response to a given probe.

As Equation 2 shows, the activation of each memory trace is an exponential function of its similarity to the probe, as

[1] This somewhat simplistic method of defining semantics was adopted for computational expedience. The semantic features lack any inherent structure (i.e., *cat* is as similar to *leopard* as it is to *democracy*). Thus, the model's performance at recalling the meanings of a large corpus of text is likely a conservative estimate of how well the model will perform when more realistic assumptions about semantic structure are incorporated in future versions of the model, e.g., by deriving features from *latent semantic analysis*.

determined by the parameter δ. The activation of a trace is also weighted by a quantity, ω_i, which represents the word's frequency of occurrence in printed text (according to the HAL frequency norms) scaled against the maximum frequency of any word in the text using Equation 3. Scaling a word's frequency in the interval [0, 1] is consistent with the notion that more common words will be encoded more often and thus be better represented in memory, and provides an approximation to encoding x memory traces for a word of frequency x. The scaling also implies that new word experiences are being encoded at a more-or-less constant rate, but with the rate of information loss due to forgetting also being more-or-less constant.

$$(2) \quad activation_i = similarity_i^{\delta}$$

$$(3) \quad \omega_i = frequency_i / \max\left[frequency_{i \in M}\right]$$

The summed activation of all of the traces in memory provides an index of a probe's *intensity*, I, as described by Equation 4. In the context of simulating the eye movements of readers, this index of global activation is assumed to be used as a heuristic indexing how soon the system will reach a stable state, allowing the initiation of saccadic programming to move the eyes to the next word (cf., the *familiarity check*—L_1—in E-Z Reader; Reichle et al., 1998, 2012).

$$(4) \quad I = \sum_{i=1}^{M} activation_i \omega_i$$

The time to generate the intensity of a given probe, $t(L_1)$, is assumed to be a proportion of the intensity, as described by Equation 5. As shown, the logarithm of the intensity, I, is used to linearly transform the intensity values into time (ms) using two free parameters, α_1 and α_2.

$$(5) \quad t(L_1) = \alpha_1 - \left[\log_{10}(I)\alpha_2\right]$$

The process of recalling lexical information uses a subset of probe features to construct a composite pattern of features, where C_j is the sum across traces of feature j in trace i multiplied by the activation of trace i, as described by Equation 6. The resulting pattern of features is the *content* of recall.

$$(6) \quad C_j = \sum_{i=1}^{M} \left(activation_i trace_{i,j} \omega_i\right)$$

The recall content typically exhibits some degree of noise because it reflects all of the information contained in memory. The noisy content pattern can be 'cleaned up' for the purpose of, for example, naming the word by normalizing the features so that the feature values span the range [0, 1]. This is done by first identifying the feature having the maximal absolute value and then dividing all of the recalled features by that value, as described by Equation 7. The resulting pattern of features, N_j, is called the *normalized content*.

$$(7) \quad N_j = C_j / \max\left[\left|C_j\right|\right]$$

The time needed to generate the normalized content is assumed to vary as a function of the maximal difference in the pre-normalized feature values [0, $|C_j|$]. The assumption is consistent with the basic intuition that smaller absolute discrepancies among the pre-normalized features will require more time to settle into a stable pattern—one that represents, for example, the pronunciation of a word with any degree of accuracy. The time required to generate the normalized content, $t(L_2)$, is thus the sum of the time required to generate the intensity, $t(L_1)$, and a term that reflects the maximum absolute feature value, max[$|C_j|$], scaled by two free parameters, α_3 and α_4.

$$(8) \quad t(L_2) = t(L_1) + \alpha_3 - \left\{\max\left[\left|C_j\right|\right]\alpha_4\right\}$$

Finally, the visual 'front end' of Über-Reader is based on principles of the Overlap model (Gomez et al., 2008) in that the visual evidence supporting the existence of a given letter in a particular location is not precise, but is instead normally distributed around some true location, as described by Equations 9 and 10. Equation 9 specifies the strength of evidence supporting letter i in position x given that it is located in some true position, μ. The degree of uncertainty is determined by the variability associated with the evidence, as determined by σ. As Equation 10 indicates, this variability increases with the absolute distance (in character spaces) between the fixation position (i.e., center of vision) and the true location of letter i, as scaled by two free parameters, β_1 and β_2. Thus, in the process of identifying a given word, the features representing the orthographic input will often reflect some degree of uncertainty or noise that will increase with the eccentricity of the word. Returning to our previous example of the word "cat," the visual input for letters "c," "a," and "t" in letter positions 1-3 would respectively be 1.0, 1.0, and 1.0 from a fixation on the word, but would drop off to 0.38, 0.35, and 0.32 from a fixation located 10 character spaces to the left of the word. There would also be uncertainty about the precise location of each letter; for example, evidence for the letter "t" in its true location would be 0.32, but would equal 0.23 for the two spatially adjacent locations and 0.09 for the next two more distant spatial locations. However, to avoid excessive interference due to low-level visual noise, any feature less than some threshold value, θ_{feature}, is not included in the probe that is used to identify words.

$$(9) \quad f(x,\mu,\sigma) = \left(1/\sigma\sqrt{2\pi}\right)e^{-(x-\mu)^2/2\sigma^2}$$

$$(10) \quad \sigma = \beta_1 + \beta_2\left|fixation - letter_i\right|$$

The time required for visual input to propagate from the eyes to the mind is delimited by the eye-mind lag. Based on Reichle and Reingold's (2013) empirical estimates the duration of this eye-mind lag, $t(V)$, is set equal to 60 ms.

The model's assumptions about saccadic programming and execution were borrowed from E-Z Reader (Reichle et al., 2012). Über-Reader includes additional assumptions related to the processing of sentences, and encoding, representing, and recalling the meaning of a text. Space limitations prohibit a full description of these assumptions here (see Reichle, 2020). In brief, a set of 17 productions (i.e., if-then statements in procedural memory that operate on the information accessed from the words) parse the sentences into phrase structures so that another set of four productions can then extract the meanings of key words. The meanings of these words are used to build a composite pattern of semantic features that is actively maintained in working memory until a sentence boundary is reached; at that time, the pattern can be encoded into long-term episodic memory. This pattern can also be recalled using Equations 1, 2, 6, and 7.

Simulating Word-Identification Tasks

This section describes the task-specific assumptions required to simulate three commonly used word-identification tasks that share a common lexical-processing component: natural reading, lexical decision, and word naming.

Two key assumptions allow patterns of eye movements during reading to be simulated. First, the intensity of a given word provides a signal to the oculomotor system that the identification of that word is imminent, causing the system to initiate the programming of a saccade to move the eyes to the next viewing location (which in most instances is the next word). The second key assumption is that the information contained in the normalized content is sufficient for further semantic and syntactic processing and, as such, causes attention to be shifted to the next word. Thus, the times required to initiate saccadic programming versus the shifting of attention are respectively given by $t(L_1)$ and $t(L_2)$. Über-Reader's distinction between intensity and content, which is inherited from MINERVA 2, corresponds to the two stages of lexical processing in E-Z Reader—the familiarity check (L_1) and the completion of lexical access (L_2), respectively.

The times required to perform other word-identification tasks reflect the mental operations involved in making decisions and the motoric operations required to execute responses. The details of these operations are not simulated here; instead, only the mean times required to make decisions, $t(D)$, and execute responses, $t(R)$, are specified. The values of these two parameters are sampled from gamma distributions with means equal to $t(D)$ and $t(R)$, respectively, and standard deviations equal to 0.22 of the means. Thus, as Equation 11 shows, the response time, RT, to name a word or to make a lexical decision about a string of letters is the sum of four components: the eye-mind lag, $t(V)$, the time required to resolve whatever lexical features are required to make a response, $t(L_2)$, the decision-making time, $t(D)$, and the response-execution time, $t(R)$.

$$RT = t(V) + t(L_2) + t(D) + t(R) \tag{11}$$

In simulating lexical decision and naming, any lexical feature exceeding the noise threshold, θ_{feature}, are considered viable in making overt responses. Features that are mutually incompatible are assumed to compete in a 'winner-take-all' manner in making those responses. For example, in generating the pronunciation for the word "cat," the values of the phoneme features (e.g., those corresponding to the phonemes /k/, /æ/, and /t/) are recalled in the normalized content. Those features and any others that exceed the noise threshold are then compared, and the most active feature in each phoneme position is selected for inclusion in the response. This method is consistent with the assumption that mutual inhibition among competing features is sufficient to dampen all but the most active features in a winner-take-all manner (see Ans et al., 1998.) The operations required to actually make the response are not simulated, but the response is scored for the purposes of assessing the model's response accuracy by calculating the proportion of features correctly recalled. Any response exceeding some criterion of accuracy, θ_{response}, is then scored as being correct.

Using the above assumptions, the lexical-decision task (LDT) is simulated by scoring the orthographic features that are recalled at time $t(L_2)$; an orthographic pattern that equals or exceeds some threshold of accuracy ($\theta_{\text{response}} = 0.9$) is scored as being a 'word' response and the response latency is the time specified by Equation 11. An orthographic pattern that does not exceed the response threshold is conversely scored as a 'non-word' response. In a similar manner, the naming task is simulated by scoring the phonological features that are recalled at time $t(L_2)$; a phonological pattern that equals or exceeds some threshold of accuracy ($\theta_{\text{response}} = 1$) is scored as having been correctly pronounced and the response latency is also the time specified by Equation 11. Finally, in simulating reading, the proportion of correctly recalled semantic features must exceed a threshold ($\theta_{\text{response}} = 0.5$) for the word to be considered "identified."

Table 1 provides a summary of the model's parameters, along with their values and short descriptions of their interpretations. The simulations reported below use the Schilling et al. (1998) corpus, which includes item-level data for lexical decision, naming, and various eye-movement measures for the same 48 target words, comprising 24 high-frequency (HF; $M = 10.32$ Log HAL) and 24 low-frequency (LF; $M = 5.93$) words. The target words are 6-9 letters in length and contain 1-4 syllables.

For the simulations of the LDT, non-words were created by randomly replacing a single letter in each of the target words to create one-letter-different non-words as used in the Schilling et al. study. The simulations of lexical decision and naming were each conducted with 100 simulated subjects.

Table 1: Über-Reader parameter values used in simulating the LDT, naming task, and sentence reading.

Parameter	Value	Description
α_1	200/87	Time to generate intensity, $t(L_1)$, intercept (ms) *Note*: LDT & naming = 200; reading = 87
α_2	50/19	Time to generate intensity, $t(L_1)$, slope (ms) *Note*: LDT & naming = 50; reading = 19
α_3	50/32	Time to generate normalized content, $t(L_2)$, intercept (ms) *Note*: LDT & naming = 50; reading = 32
α_4	60/24	Time to generate normalized content, $t(L_2)$, slope (ms) *Note*: LDT & naming = 60; reading = 24
β_1	0.05	Letter-position uncertainty gradient intercept (character position)
β_2	0.05	Letter-position uncertainty gradient slope (character position)
δ	17	Similarity gradient for trace activation
$p_{semantic}$	0.02	Probability of semantic feature being active
$\theta_{similarity}$	0.9	Minimal probe-trace similarity for trace activation
$\theta_{feature}$	0.1	Feature activation threshold
$\theta_{response}$	0.9/1/0.5	Goodness-of-response threshold *Note*: LDT = 0.9; naming = 1; reading = 0.5
$t(D)$	100	Mean decision time (ms)
$t(R)$	100	Mean response time (ms)
$t(V)$	60	Eye-mind lag (ms)

Simulation Results

Lexical Decision

The results of the simulation of the LDT are presented in Figure 2. As shown, Über-Reader accounted for a good proportion of the item-level variance in lexical-decision RTs (R^2 = 0.58). The simulated data also clearly reproduced the frequency effect on lexical decision latencies. The model was 100% accurate at discriminating the 48 Schilling et al. target words from one-letter-different non-words. In contrast, Schilling et al. reported accuracy of 97% for HF words and 89% for LF words.

Word Naming

The results of the simulation of word naming are presented in Figure 3. Über-Reader was again found to account for a good proportion of the item-level variance in naming latencies (R^2 = 0.52) and reproduced the frequency effect on naming latencies. The model was also 100% accurate at naming the 48 Schilling et al. target words. In contrast, Schilling et al. reported accuracy of 98% for HF words and 97% for LF words.

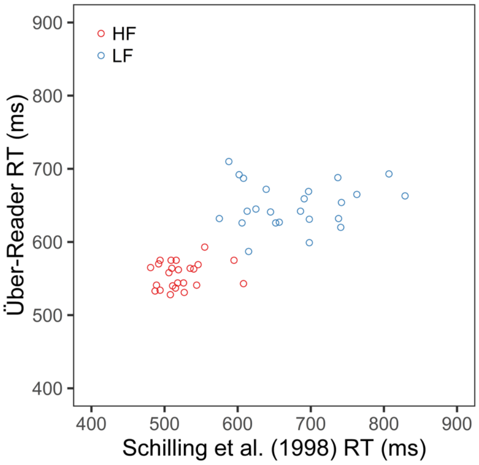

Figure 2: Item-level mean LDT RTs for the Schilling et al. (1998) targets vs. Über-Reader simulated data.

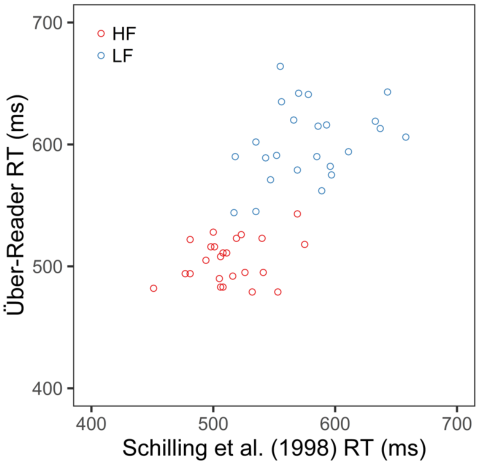

Figure 3: Item-level mean naming latencies for the Schilling et al. (1998) targets vs. Über-Reader simulated data.

Sentence Reading

Figure 4 shows six observed (*Obs*) and simulated (*Sim*) eye-movement measures as a function of word frequency, with all calculated using first-pass eye movements (i.e., excluding fixations following inter-word regressions back to earlier parts of the text). Panel A shows: (1) *first-fixation duration* (*FFD*), the duration of the first of possibly several fixations on a word; (2) *single-fixation duration* (*SFD*), the duration of the fixation on a word that is fixated exactly once; and (3) *gaze duration* (*GD*), the sum of all first-pass fixations. Panel B shows the: (4) probability of fixating a word exactly once (*Pr1*); (5) probability of fixating a word two or more times

(*Pr2+*); and (6) probability of skipping a word (*PrS*). As shown, the model provided good fits to all of these measures; for example, as word frequency increases, both the probability of fixating a word and the durations of those fixations tends to decrease.

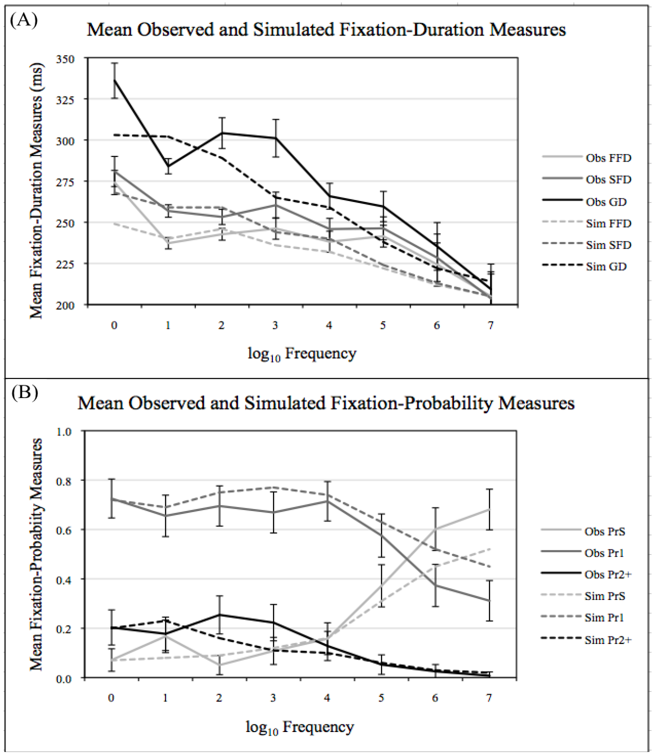

Figure 4: Mean observed and simulated: (A) fixation-duration and (B) fixation-probability measures for the Schilling et al. (1998) sentences.

Figure 5 shows three other simulated "benchmark" eye-movement findings, all of which are shown as a function of both word length and initial fixation landing position (with 0 being the blank space to the left of a word). Panel A shows the simulated fixation landing-site distributions, which are approximately normal in shape due to the fact that the eyes are directed towards the centers of words but often miss their mark due to both systematic and random error (McConkie et al., 1988). Panel B shows the probabilities of making a refixation, which are U-shaped but asymmetrical due to the fact that words are most likely to be refixated following an initial fixation near the beginning of a word (McConkie et al., 1989). Finally, Panel C shows SFDs as a function of their location; as has been observed, SFDs are longer for fixations near the centers than ends of words, resulting in *inverted optimal-viewing position* (*IOVP*) effects (Vitu et al., 2001). These simulated results are important because they demonstrate that the direct simulation of the sequence of processes required for word identification operates sufficiently quickly to yield the trade-offs that occur between lexical processes and the programming and execution of saccades that manifest themselves in the empirical phenomena depicted in Figure 5.

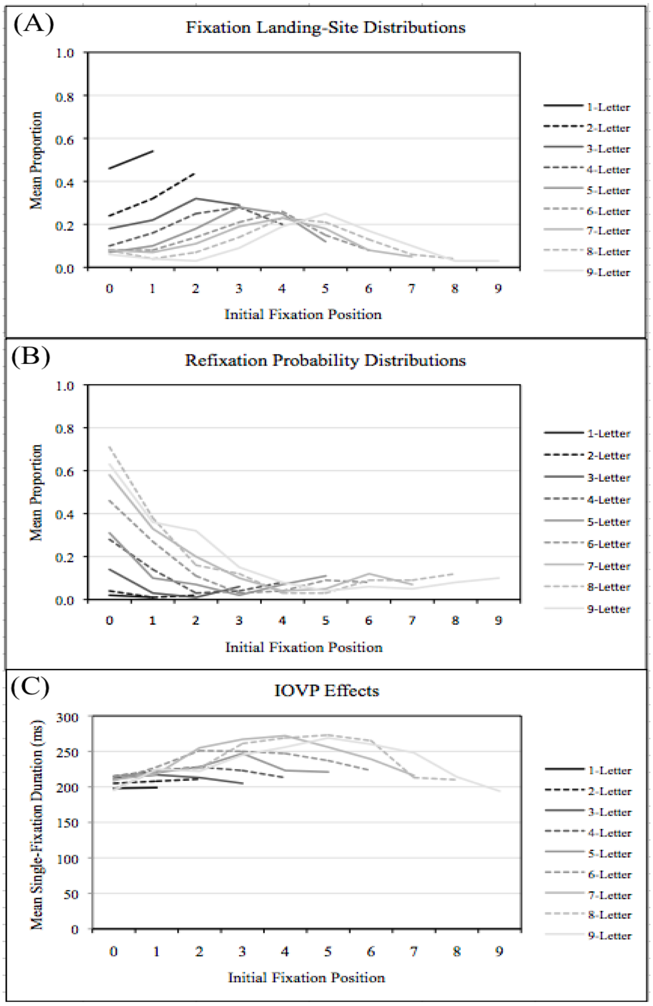

Figure 5: Three simulated "benchmark" eye-movement phenomena: (A) fixation landing-site distributions; (B) refixation-probability distributions; and (C) IOVP effects.

Discussion

The present work represents the first attempt to simulate three tasks that have been extensively used to study reading using the instance-based, word-identification component of Über-Reader. Nevertheless, Über-Reader accounts for an impressive proportion of the item-level variance in the Schilling et al. (1998) corpus for all three tasks.

The simulations of sentence reading produced comparable fits of the eye-movement data to E-Z Reader, and Über-Reader replicated several benchmark eye-movement effects. This was important to verify because, despite the assumptions about saccade planning and execution being inherited from its predecessor, Über-Reader makes additional assumptions about the processes and time-course of word identification and directly simulates the retrieval of lexical content from long-term memory. These simulations therefore demonstrate that an instance-based model of word identification can successfully be incorporated into the architecture of an existing model of eye-movement control and account for patterns of eye movements in reading. Furthermore, the distinction between intensity and content in Über-Reader, and the interface of this resonance-retrieval process with saccadic planning and attention shifting, addresses a key limitation of E-Z Reader in which the equivalent distinction between the two stages of lexical processing (*L_1* and *L_2*) that is central to the model's account remains theoretically underspecified.

This demonstration must be interpreted with some caution, however. The fact that different values of the lexical-processing parameters provided optimal fits to the data from tasks that were performed by the same set of participants may reflect limitations in how the parameter values were selected (via grid-searches of the parameter spaces using relatively few statistical subjects due to the fact that each required approximately 2.75 minutes on a 2.3 GHz Intel Xeon W processor). However, it is perhaps unsurprising that the parameter values differed between isolated word-identification tasks and sentence reading. In contrast to lexical decision or naming, identifying words in sentences involves the contribution of top-down information derived from the context, even in unconstraining sentences, which likely makes markedly different demands on the word-identification component common to the three tasks. The next stage of this project will refine the sentence- and discourse-processing assumptions of the model to simulate higher-level contextual effects on word identification and confirm these task differences.

Additional assumptions may also be required to capture RT distributions. One notable feature of the LDT and naming simulations was that the model was 100% accurate for the 48 Schilling et al. target words. While human performance was effectively at ceiling in the naming task for these items, the model substantially outperformed participants in lexical decision accuracy for the LF items. One possible solution to this discrepancy would be to incorporate an evidence accumulator for making the word/nonword decision in the LDT. Additional modelling work that is not reported here tested a version of the model that included such an evidence accumulator but resulted in unacceptable trade-offs between word/nonword discrimination performance and fits to the item-level RT data. Other possibilities would be the addition of noise to the decision and/or response execution stages of the LDT, or trial-by-trial adjustments to the response threshold parameter on the basis of previous trial response accuracy.

Acknowledgements

This research was supported under Australian Research Council's Discovery Projects funding scheme (project number DP190100719).

References

Andrews, S. & Reichle, E. D. (2019). The cognitive architecture of reading: The organziation of an acquired skill. In P. Hagoort (Ed.), *Human language: From genes and brains to behavior* (pp. 51-66). MIT Press.

Ans, B., Carbonnel, S., & Valdois, S. (1998). A connectionist multiple-trace memory model for polysyllabic word reading. *Psychological Review, 105*, 687-723.

Balota, D. A., Yap, M. J., Hutchison, K. A., Cortese, M. J., Kessler, B., Loftis, B., Neely, J. H., Nelson, D. L., Simpson, G. B., & Treiman, R. (2007). The English Lexicon Project. *Behavior Research Methods, 39*, 445-459.

Dougherty, M. R. P., Gettys, C. F., & Ogden, E. E. (1999). MINERVA-DM: A memory processes model for judgments of likelihood. *Psychological Review, 106*, 180-209.

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

Gomez, P., Ratcliff, R., & Perea, M. (2008). The overlap model: A model of letter position coding. *Psychological Review, 115*, 577-600.

Hintzman, D. L. (1984). MINERVA 2: A simulation model of human memory. *Behavior Research Methods, Instruments, & Computers, 16*, 96-101.

Huey, E. B. (1908). *The psychology and pedagogy of reading*. New York, NY, USA: Macmillan.

Kintsch, W. (1998). *Comprehension: A paradigm for cognition*. Cambridge University Press.

McConkie, G. W., Kerr, P. W., Reddix, M. D., & Zola, D. (1988). Eye movement control during reading: I. The location of initial eye fixations in words. *Vision Research, 28*, 1107-1118.

McConkie, G. W., Kerr, P. W., Reddix, M. D., Zola, D., Jacobs, A. M. (1989). Eye movement control during reading: II. Frequency of refixating a word. *Perception & Psychophysics*, *46*, 245-253.

Reichle, E. D. (2020). *Computational models of reading: A handbook*. Oxford University Press.

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

Reichle, E. D., Pollatsek, A., & Rayner, K. (2012). Using E-Z Reader to simulate eye movements in nonreading tasks: A unified framework for understanding the eye-mind link. *Psychological Review, 119*, 155-185.

Reichle, E. D., & Reingold, E. M. (2013). Neurophysiological constraints on the eye-mind link. *Frontiers in Human Neuroscience, 7*, 1-6.

Schilling, H. E. H., Rayner, K., & Chumbley, J. I. (1998). Comparing naming, lexical decision, and eye fixation times: Word frequency effects and individual differences. *Memory and Cognition, 26*, 1270-1281.

Van Dyke, J. A., & Lewis, R. L. (2003). Distinguishing effects of structure and decay on attachment and repair: A cue-based parsing account of recovery from misanalyzed ambiguities. *Journal of Memory and Language, 49*, 285-316.

Vitu, F., McConkie, G. W., Kerr, P., & O'Regan, J. K. (2001). Fixation location effects on fixation durations during reading: An inverted optimal viewing position effect. *Vision Research*, *41*, 3513-3533.