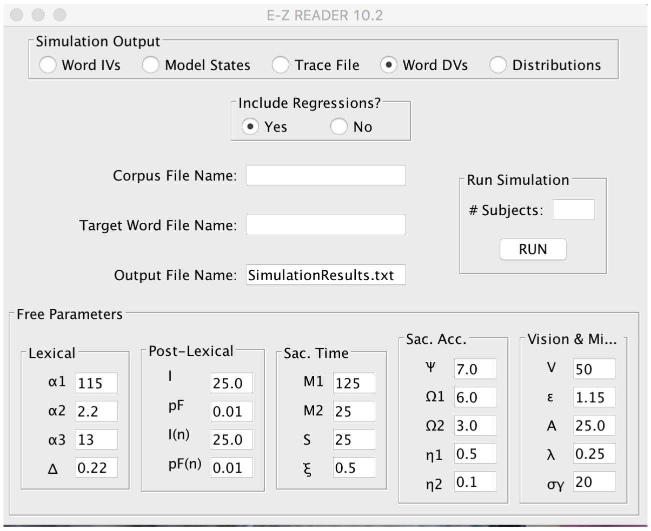
Running E-Z Reader 10.2 Simulations

The program for running simulations using the E-Z Reader 10.2 model was written in Java, version 1.8. Both the executable (*.jar*) version of the program and the source code (i.e., *.java* classes) are available from my website (www.erikdreichle.com) and upon request (erik.reichle@mq.edu.au). The first part of these instructions describes how to run simulations using the executable program and the Schilling, Rayner, and Chumbley (1998) sentence corpus. The second part describes how to run simulations using your own sentence corpus.

1. Running Simulations

You will need three files to run E-Z Reader simulations: (1) the program file containing the model (*E-Z Reader 10.jar*); (2) a file containing the sentences that will be used in the simulation (e.g., *SRC98Corpus.txt*); and (3) a file used to identify specific target words of interest (e.g., *SRC98Targets.txt*). To run a simulation, first download these files to your computer desktop or a common folder and then double-click on the program file. This should open a graphic-user interface (GUI; see Fig. 1, below) with buttons and text fields that can be selected or modified for running different types of simulations. Here is a brief explanation of the GUI.


E-Z READER 10.2

Simulation Output
○ Word IVs ○ Model States ○ Trace File ● Word DVs ○ Distributions

Include Regressions?
● Yes ○ No

Corpus File Name:
Target Word File Name:
Output File Name: SimulationResults.txt

Run Simulation
Subjects:
RUN

Free Parameters

Lexical
α1 115
α2 2.2
α3 13
Δ 0.22

Post-Lexical
I 25.0
pF 0.01
I(n) 25.0
pF(n) 0.01

Sac. Time
M1 125
M2 25
S 25
ξ 0.5

Sac. Acc.
Ψ 7.0
Ω1 6.0
Ω2 3.0
η1 0.5
η2 0.1

Vision & Mi...
V 50
ε 1.15
A 25.0
λ 0.25
σγ 20

Figure 1. E-Z Reader GUI.

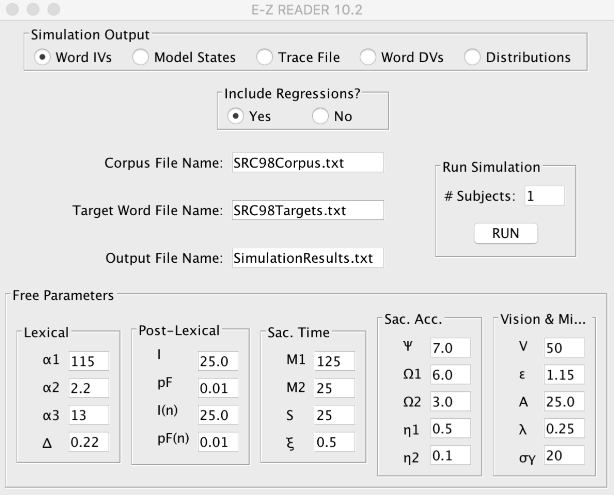
The only information that must be entered into the GUI before you can start running a simulation is the following:

(1.) *Corpus File Name* – Enter the name of the file containing the sentences that will be used in the simulation. The example file *SRC98Corpus.txt* contains the 48 sentences used by Schilling et al. (1998) in their eye-movement experiment and subsequently used to evaluate different versions of the E-Z Reader model.

(2.) # *Subjects* – Enter the number of statistical subjects (1-10,000) that will be used in completing the simulation.

(3.) *RUN* – Press this button to start the simulation. The length of time required to complete a simulation will depend upon the speed of your computer and other variables (e.g., the number of statistical subjects).

Figure 2 (below) shows an example of what the GUI will look like when information has been entered.



● Word IVs ○ Model States ○ Trace File ○ Word DVs ○ Distributions

● Yes ○ No

Corpus File Name: SRC98Corpus.txt

Target Word File Name: SRC98Targets.txt

Output File Name: SimulationResults.txt

Subjects: 1

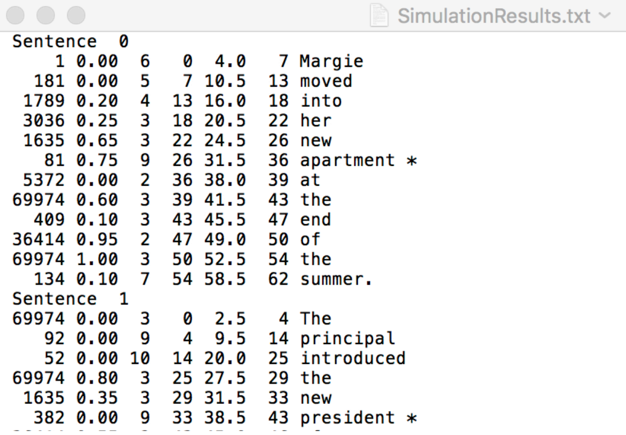
Lexical		Post-Lexical		Sac. Time		Sac. Acc.		Vision & Mi...	
α1	115	I	25.0	M1	125	Ψ	7.0	V	50
α2	2.2	pF	0.01	M2	25	Ω1	6.0	ε	1.15
α3	13	I(n)	25.0	S	25	Ω2	3.0	A	25.0
Δ	0.22	pF(n)	0.01	ξ	0.5	η1	0.5	λ	0.25
						η2	0.1	σγ	20

Figure 2. E-Z Reader GUI with input files and the number of statistical subjects entered.

All of the other buttons and text fields are set to their default values, but can be modified as necessary. The text field labeled *Output File Name* names the file to which simulation results will be written (default name: *Simulation Results.txt*). This file will appear in the same location as the program file after the program runs to completion. (Note that changing the *.txt* file extension to *.xls* will cause the output to be formatted for a space-delimited Microsoft EXCEL file, making it easier to analyze.) The text fields in the box labeled *Parameter Values* contain E-Z Reader's default parameter values. Two things are important to remember about these values. First, the values of "I(n)" and "pF(n)" are used to set the values of I and p_F, respectfully, for specific target words (i.e., as specified by, e.g., *SRC98Targets.txt*). Second, the parameter that controls the gamma distribution variability, $\sigma_\gamma = 20$, is set equal to a value that generates gamma distributions having standard deviations equal to 0.22 of their means. [For more information about the gamma distribution function that is used in the E-Z Reader program, see Press, Teukolsky, Vetterling, & Flannery (1992). *Numerical Recipes in C: The Art of Scientific Computing*. New York: Cambridge University Press.]

Finally, the buttons in the box labeled *Simulation Output* can be selected to write different types of simulation results to the output file:

(1.) *Word IVs* – Selecting this button will output all of the independent variables associated with all of the words in the sentence file that are calculated by the program prior to executing a simulation (e.g., each word's *optimal viewing position*, or *OVP*). It's a good idea to run this simulation prior to completing any others to ensure that the sentence file has been formatted correctly. (It's also a good idea to use a single statistical subject to avoid generating an extremely large text file.) Figure 3 shows an example of the first part of the output from this type of simulation and an explanation of what it means:

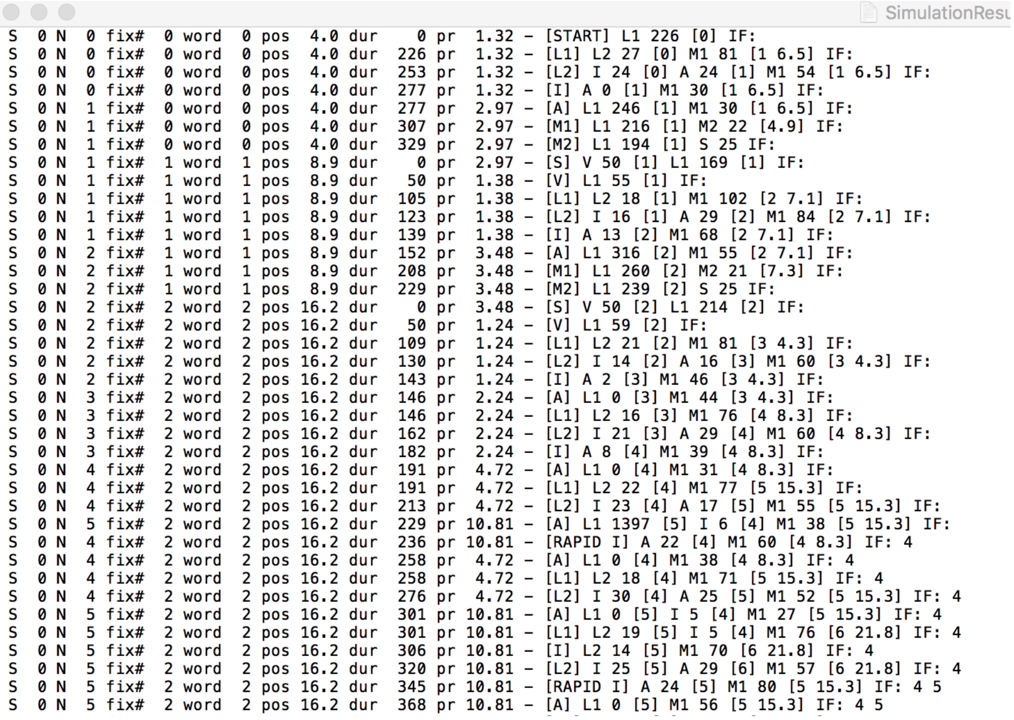


```
Sentence  0
    1 0.00  6   0  4.0   7 Margie
  181 0.00  5   7 10.5  13 moved
 1789 0.20  4  13 16.0  18 into
 3036 0.25  3  18 20.5  22 her
 1635 0.65  3  22 24.5  26 new
   81 0.75  9  26 31.5  36 apartment *
 5372 0.00  2  36 38.0  39 at
69974 0.60  3  39 41.5  43 the
  409 0.10  3  43 45.5  47 end
36414 0.95  2  47 49.0  50 of
69974 1.00  3  50 52.5  54 the
  134 0.10  7  54 58.5  62 summer.
Sentence  1
69974 0.00  3   0  2.5   4 The
   92 0.00  9   4  9.5  14 principal
   52 0.00 10  14 20.0  25 introduced
69974 0.80  3  25 27.5  29 the
 1635 0.35  3  29 31.5  33 new
  382 0.00  9  33 38.5  43 president *
```

Figure 3. Example output from "Word IVs" simulation.

The above example shows the first sentence (i.e., "Sentence: 0") and part of the second (i.e., "Sentence: 1"). Following Java conventions, sentences and words are always numbered starting from 0, so that a set of *N* sentences/words will be numbered from 0 to *N*-1. Each row shows the following information for a given word: (1) its frequency of occurrence in printed text (e.g., Francis & Kucera, 1982); (2) its cloze predictability (Taylor, 1953); (3) its length (i.e., number of letters); (4) the cumulative character position of the space immediately to the left of the word; (4) the cumulative character position of the center of the word (i.e., its OVP); (5) the cumulative character position of the right side the last character in a word; (6) the actual word; and (7) an asterisk marking target words.

(2.) *Model States* – Selecting this button will cause the model to write out all of the internal states that it progresses through as it "reads" the sentences. This type of output is useful for seeing how the model works, and can sometimes be useful for figuring out exactly *why* the model makes certain predictions. Because the output files are very large (each word that is processed might cause the model to go through more than 10 states), it is a good idea to use only a very small number of subjects when running this type of simulation. Figure 4 shows an example of the output and an explanation of what it means:



```
S  0 N  0 fix#  0 word  0 pos  4.0 dur    0 pr  1.32 - [START] L1 226 [0] IF:
S  0 N  0 fix#  0 word  0 pos  4.0 dur  226 pr  1.32 - [L1] L2 27 [0] M1 81 [1 6.5] IF:
S  0 N  0 fix#  0 word  0 pos  4.0 dur  253 pr  1.32 - [L2] I 24 [0] A 24 [1] M1 54 [1 6.5] IF:
S  0 N  0 fix#  0 word  0 pos  4.0 dur  277 pr  1.32 - [I] A 0 [1] M1 30 [1 6.5] IF:
S  0 N  1 fix#  0 word  0 pos  4.0 dur  277 pr  2.97 - [A] L1 246 [1] M1 30 [1 6.5] IF:
S  0 N  1 fix#  0 word  0 pos  4.0 dur  307 pr  2.97 - [M1] L1 216 [1] M2 22 [4.9] IF:
S  0 N  1 fix#  0 word  0 pos  4.0 dur  329 pr  2.97 - [M2] L1 194 [1] S 25 IF:
S  0 N  1 fix#  1 word  1 pos  8.9 dur    0 pr  2.97 - [S] V 50 [1] L1 169 [1] IF:
S  0 N  1 fix#  1 word  1 pos  8.9 dur   50 pr  1.38 - [V] L1 55 [1] IF:
S  0 N  1 fix#  1 word  1 pos  8.9 dur  105 pr  1.38 - [L1] L2 18 [1] M1 102 [2 7.1] IF:
S  0 N  1 fix#  1 word  1 pos  8.9 dur  123 pr  1.38 - [L2] I 16 [1] A 29 [2] M1 84 [2 7.1] IF:
S  0 N  1 fix#  1 word  1 pos  8.9 dur  139 pr  1.38 - [I] A 13 [2] M1 68 [2 7.1] IF:
S  0 N  2 fix#  1 word  1 pos  8.9 dur  152 pr  3.48 - [A] L1 316 [2] M1 55 [2 7.1] IF:
S  0 N  2 fix#  1 word  1 pos  8.9 dur  208 pr  3.48 - [M1] L1 260 [2] M2 21 [7.3] IF:
S  0 N  2 fix#  1 word  1 pos  8.9 dur  229 pr  3.48 - [M2] L1 239 [2] S 25 IF:
S  0 N  2 fix#  2 word  2 pos 16.2 dur    0 pr  3.48 - [S] V 50 [2] L1 214 [2] IF:
S  0 N  2 fix#  2 word  2 pos 16.2 dur   50 pr  1.24 - [V] L1 59 [2] IF:
S  0 N  2 fix#  2 word  2 pos 16.2 dur  109 pr  1.24 - [L1] L2 21 [2] M1 81 [3 4.3] IF:
S  0 N  2 fix#  2 word  2 pos 16.2 dur  130 pr  1.24 - [L2] I 14 [2] A 16 [3] M1 60 [3 4.3] IF:
S  0 N  2 fix#  2 word  2 pos 16.2 dur  143 pr  1.24 - [I] A 2 [3] M1 46 [3 4.3] IF:
S  0 N  3 fix#  2 word  2 pos 16.2 dur  146 pr  2.24 - [A] L1 0 [3] M1 44 [3 4.3] IF:
S  0 N  3 fix#  2 word  2 pos 16.2 dur  146 pr  2.24 - [L1] L2 16 [3] M1 76 [4 8.3] IF:
S  0 N  3 fix#  2 word  2 pos 16.2 dur  162 pr  2.24 - [L2] I 21 [3] A 29 [4] M1 60 [4 8.3] IF:
S  0 N  3 fix#  2 word  2 pos 16.2 dur  182 pr  2.24 - [I] A 8 [4] M1 39 [4 8.3] IF:
S  0 N  4 fix#  2 word  2 pos 16.2 dur  191 pr  4.72 - [A] L1 0 [4] M1 31 [4 8.3] IF:
S  0 N  4 fix#  2 word  2 pos 16.2 dur  191 pr  4.72 - [L1] L2 22 [4] M1 77 [5 15.3] IF:
S  0 N  4 fix#  2 word  2 pos 16.2 dur  213 pr  4.72 - [L2] I 23 [4] A 17 [5] M1 55 [5 15.3] IF:
S  0 N  5 fix#  2 word  2 pos 16.2 dur  229 pr 10.81 - [A] L1 1397 [5] I 6 [4] M1 38 [5 15.3] IF:
S  0 N  4 fix#  2 word  2 pos 16.2 dur  236 pr 10.81 - [RAPID I] A 22 [4] M1 60 [4 8.3] IF: 4
S  0 N  4 fix#  2 word  2 pos 16.2 dur  258 pr  4.72 - [A] L1 0 [4] M1 38 [4 8.3] IF: 4
S  0 N  4 fix#  2 word  2 pos 16.2 dur  258 pr  4.72 - [L1] L2 18 [4] M1 71 [5 15.3] IF: 4
S  0 N  4 fix#  2 word  2 pos 16.2 dur  276 pr  4.72 - [L2] I 30 [4] A 25 [5] M1 52 [5 15.3] IF: 4
S  0 N  5 fix#  2 word  2 pos 16.2 dur  301 pr 10.81 - [A] L1 0 [5] I 5 [4] M1 27 [5 15.3] IF: 4
S  0 N  5 fix#  2 word  2 pos 16.2 dur  301 pr 10.81 - [L1] L2 19 [5] I 5 [4] M1 76 [6 21.8] IF: 4
S  0 N  5 fix#  2 word  2 pos 16.2 dur  306 pr 10.81 - [I] L2 14 [5] M1 70 [6 21.8] IF: 4
S  0 N  5 fix#  2 word  2 pos 16.2 dur  320 pr 10.81 - [L2] I 25 [5] A 29 [6] M1 57 [6 21.8] IF: 4
S  0 N  5 fix#  2 word  2 pos 16.2 dur  345 pr 10.81 - [RAPID I] A 24 [5] M1 80 [5 15.3] IF: 4 5
S  0 N  5 fix#  2 word  2 pos 16.2 dur  368 pr 10.81 - [A] L1 0 [5] M1 56 [5 15.3] IF: 4 5
```

Figure 4. Example output from "Model States" simulation.

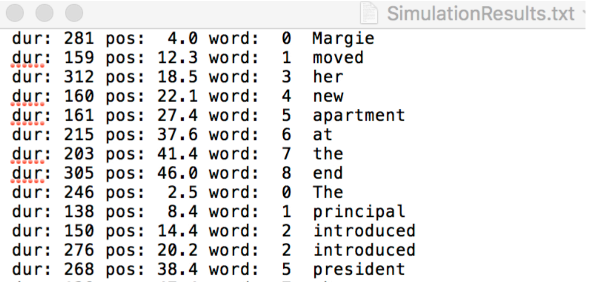
The above example shows consecutive model states, displayed one per row. Within each row, the following information is provided (from left to right): (1) S - the current sentence being read (e.g., the first sentence); (2) N - where attention is located (i.e., the word being processed); (3) fix# - the fixation number; (4) word - the word being fixated; (5) pos - the cumulative character position of the current fixation location; (6) dur - the duration of the current fixation; (7) pr - the current rate of lexical processing; (8) a model state (as described below); (9) a list of on-going processes and information associated with those processes (also as described below); and (10) IF: - a list of words for which integration has failed (e.g., as shown in the last row, integration has failed for words 4 and 5).

The on-going processes listed in (9) above are: (i) V, pre-attentive vision; (ii) L1, the first stage of lexical processing (i.e., the familiarity check); (iii) L2, the second stage of lexical processing (i.e., lexical access); (iv)) I, post-lexical integration; (v) A, attention shift; (vi) M1, labile saccadic programming; (vii) M2, non-labile saccadic programming; and (viii) S, the saccade. The number immediately to the right of these labels indicates the amount of time remaining for a given process. For example, in the first row of Figure 4, "L1 226 [0]" indicates that the familiarity check (L1) will require 226 ms to complete for word 0 (i.e., the first word in the sentence). The other state labels and associated values are interpreted similarly, but for A, the number in the brackets indicates the word towards which attention is being shifted; e.g., in the third row of Figure 4, "A 24 [1]" indicates that attention will require 24 ms to shift, and that it will be shifting towards word 1. For

M1, the two numbers in brackets respectively indicate the word being targeted by the saccade and the intended saccade length. And for M2, the number in brackets indicates the saccade length after both random and systematic error have been added to the intended length. For example, in the second line, the duration of M1 is 81 ms, the impending saccade will be being directed towards the center of word 1 (i.e., its OVP), with an intended saccade length of 6.5 character spaces. However, as line 6 shows, the actual saccade length is only 4.9 character spaces, which as line 8 shows, the eyes then move from the OVP of word 0 (i.e., cumulative character position 4.0) to cumulative character position 8.9 in word 1.

The model states listed in (8) above are: (i) [START] - the model has started reading a sentence, the eyes are on the OVP of word 0 and L1 is initiated; (ii) [L1] - L1 has completed, L2 and M1 are initiated; (iii) [L2] - L2 has completed, A and I are initiated; (iv) [I] - I has completed; (v) [A] - A has completed, L1 is initiated; (vi) [M1] - M1 has completed, M2 is initiated; (vii) [M2] - M2 has completed, S is initiated; [S] has completed, V is initiated; (viii) [V] - V has completed, the rate of lexical processing is adjusted; and (ix) [RAPID I] - I has failed (i.e., rapid integration failure). These states are labeled in the source code for the model (see the values of the variable labeled "state" in *Model.java*.) For a detailed discussion of the model states and how state transitions occur in E-Z Reader, see Reichle et al. (1998).

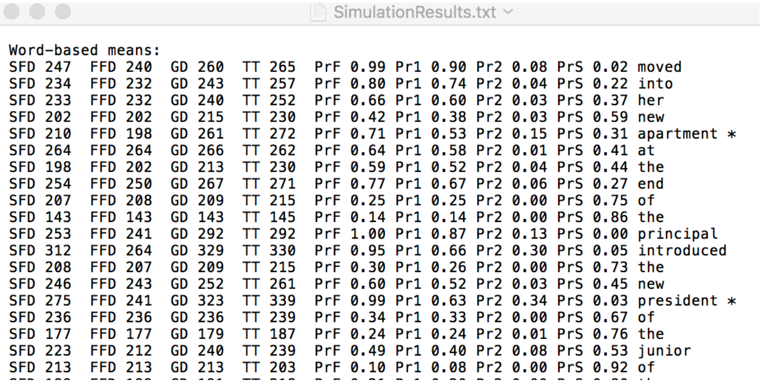
(3.) *Trace* – Selecting this button will result in the model generating a trace file that is similar to those that are generated by eye-trackers experiments using human participants. Figure 5 shows the "trace file" output, with each line containing the following information about a given fixation: (1) its duration; (2) its position; (3) the word being fixated; and (4) the identity of that word.



```
dur: 281 pos:  4.0 word:  0  Margie
dur: 159 pos: 12.3 word:  1  moved
dur: 312 pos: 18.5 word:  3  her
dur: 160 pos: 22.1 word:  4  new
dur: 161 pos: 27.4 word:  5  apartment
dur: 215 pos: 37.6 word:  6  at
dur: 203 pos: 41.4 word:  7  the
dur: 305 pos: 46.0 word:  8  end
dur: 246 pos:  2.5 word:  0  The
dur: 138 pos:  8.4 word:  1  principal
dur: 150 pos: 14.4 word:  2  introduced
dur: 276 pos: 20.2 word:  2  introduced
dur: 268 pos: 38.4 word:  5  president
```

Figure 5. Example output of "Trace File" simulation.

(4.) *Word DVs* – This output will probably be most useful for running simulations. Selecting this button will generate a number of the standard dependent measures (e.g., mean gaze durations, etc.) for each word in the sentence file. With this type of simulation, it is advisable to use a large number of subjects (e.g., 1,000) to obtain stable simulation results. Also, the predicted results for the first and last words in each sentence are not included in the output because the model: (1) starts processing the first word from its middle and with no parafoveal preview, and (2) halts abruptly (regardless of whatever is happening) when post-lexical integration of the last word has completed. (For those reasons, the dependent values of the first and last words are never included in our analyses; see Reichle et al., 1998). Figures 6-9 provide examples of the simulation output and an explanation of what it means:

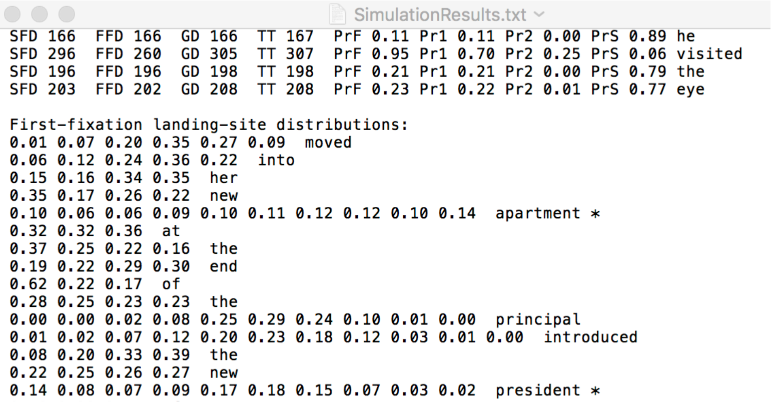


```
Word-based means:
SFD 247  FFD 240  GD 260  TT 265  PrF 0.99 Pr1 0.90 Pr2 0.08 PrS 0.02 moved
SFD 234  FFD 232  GD 243  TT 257  PrF 0.80 Pr1 0.74 Pr2 0.04 PrS 0.22 into
SFD 233  FFD 232  GD 240  TT 252  PrF 0.66 Pr1 0.60 Pr2 0.03 PrS 0.37 her
SFD 202  FFD 202  GD 215  TT 230  PrF 0.42 Pr1 0.38 Pr2 0.03 PrS 0.59 new
SFD 210  FFD 198  GD 261  TT 272  PrF 0.71 Pr1 0.53 Pr2 0.15 PrS 0.31 apartment *
SFD 264  FFD 264  GD 266  TT 262  PrF 0.64 Pr1 0.58 Pr2 0.01 PrS 0.41 at
SFD 198  FFD 202  GD 213  TT 230  PrF 0.59 Pr1 0.52 Pr2 0.04 PrS 0.44 the
SFD 254  FFD 250  GD 267  TT 271  PrF 0.77 Pr1 0.67 Pr2 0.06 PrS 0.27 end
SFD 207  FFD 208  GD 209  TT 215  PrF 0.25 Pr1 0.25 Pr2 0.00 PrS 0.75 of
SFD 143  FFD 143  GD 143  TT 145  PrF 0.14 Pr1 0.14 Pr2 0.00 PrS 0.86 the
SFD 253  FFD 241  GD 292  TT 292  PrF 1.00 Pr1 0.87 Pr2 0.13 PrS 0.00 principal
SFD 312  FFD 264  GD 329  TT 330  PrF 0.95 Pr1 0.66 Pr2 0.30 PrS 0.05 introduced
SFD 208  FFD 207  GD 209  TT 215  PrF 0.30 Pr1 0.26 Pr2 0.00 PrS 0.73 the
SFD 246  FFD 243  GD 252  TT 261  PrF 0.60 Pr1 0.52 Pr2 0.03 PrS 0.45 new
SFD 275  FFD 241  GD 323  TT 339  PrF 0.99 Pr1 0.63 Pr2 0.34 PrS 0.03 president *
SFD 236  FFD 236  GD 236  TT 239  PrF 0.34 Pr1 0.33 Pr2 0.00 PrS 0.67 of
SFD 177  FFD 177  GD 179  TT 187  PrF 0.24 Pr1 0.24 Pr2 0.01 PrS 0.76 the
SFD 223  FFD 212  GD 240  TT 239  PrF 0.49 Pr1 0.40 Pr2 0.08 PrS 0.53 junior
SFD 213  FFD 213  GD 213  TT 203  PrF 0.10 Pr1 0.08 Pr2 0.00 PrS 0.92 of
```

Figure 6. First example of output from "Word DVs" simulation, showing mean word-based dependent measures for each word.

The top part of the output file contains several mean dependent measures for each word in the sentence corpus: (1) single-fixation duration (SFD); (2) first-fixation duration (FFD); (3) gaze duration (GD); (4) total time (TT); (5) fixation probability (PrF); (6) probability of making exactly one fixation (Pr1); (7) probability of making two or more fixations (Pr2); and (8) probability of skipping (PrS). The asterisks indicate the target words that are specified by the target word input file (e.g., *SRC98Targets.txt*).

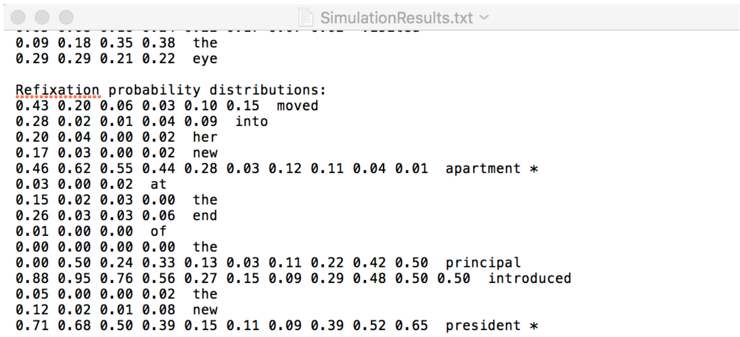
As Figure 7 shows, the next part of the output file contains the first-fixation landing-site distributions for each word:



```
SFD 166  FFD 166  GD 166  TT 167  PrF 0.11 Pr1 0.11 Pr2 0.00 PrS 0.89 he
SFD 296  FFD 260  GD 305  TT 307  PrF 0.95 Pr1 0.70 Pr2 0.25 PrS 0.06 visited
SFD 196  FFD 196  GD 198  TT 198  PrF 0.21 Pr1 0.21 Pr2 0.00 PrS 0.79 the
SFD 203  FFD 202  GD 208  TT 208  PrF 0.23 Pr1 0.22 Pr2 0.01 PrS 0.77 eye

First-fixation landing-site distributions:
0.01 0.07 0.20 0.35 0.27 0.09  moved
0.06 0.12 0.24 0.36 0.22  into
0.15 0.16 0.34 0.35  her
0.35 0.17 0.26 0.22  new
0.10 0.06 0.06 0.09 0.10 0.11 0.12 0.12 0.10 0.14  apartment *
0.32 0.32 0.36  at
0.37 0.25 0.22 0.16  the
0.19 0.22 0.29 0.30  end
0.62 0.22 0.17  of
0.28 0.25 0.23 0.23  the
0.00 0.00 0.02 0.08 0.25 0.29 0.24 0.10 0.01 0.00  principal
0.01 0.02 0.07 0.12 0.20 0.23 0.18 0.12 0.03 0.01 0.00  introduced
0.08 0.20 0.33 0.39  the
0.22 0.25 0.26 0.27  new
0.14 0.08 0.07 0.09 0.17 0.18 0.15 0.07 0.03 0.02  president *
```

Figure 7 Second example of output from "Word DVs" simulation, showing first-fixation landing-site distributions for each word.

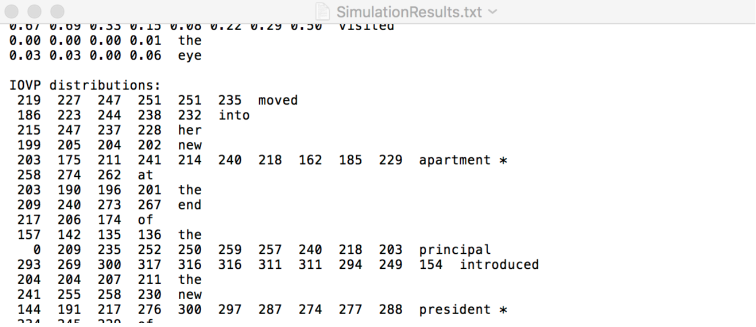
As Figure 8 shows, the next part of the output file contains the refixation-probability distributions (i.e., probability of refixating from each initial fixation position) for each word:



```
0.09 0.18 0.35 0.38  the
0.29 0.29 0.21 0.22  eye

Refixation probability distributions:
0.43 0.20 0.06 0.03 0.10 0.15  moved
0.28 0.02 0.01 0.04 0.09  into
0.20 0.04 0.00 0.02  her
0.17 0.03 0.00 0.02  new
0.46 0.62 0.55 0.44 0.28 0.03 0.12 0.11 0.04 0.01  apartment *
0.03 0.00 0.02  at
0.15 0.02 0.03 0.00  the
0.26 0.03 0.03 0.06  end
0.01 0.00 0.00  of
0.00 0.00 0.00 0.00  the
0.00 0.50 0.24 0.33 0.13 0.03 0.11 0.22 0.42 0.50  principal
0.88 0.95 0.76 0.56 0.27 0.15 0.09 0.29 0.48 0.50 0.50  introduced
0.05 0.00 0.00 0.02  the
0.12 0.02 0.01 0.08  new
0.71 0.68 0.50 0.39 0.15 0.11 0.09 0.39 0.52 0.65  president *
```

Figure 8. Third example of output from “Word DVs” simulation, showing refixation-probability distributions for each word.

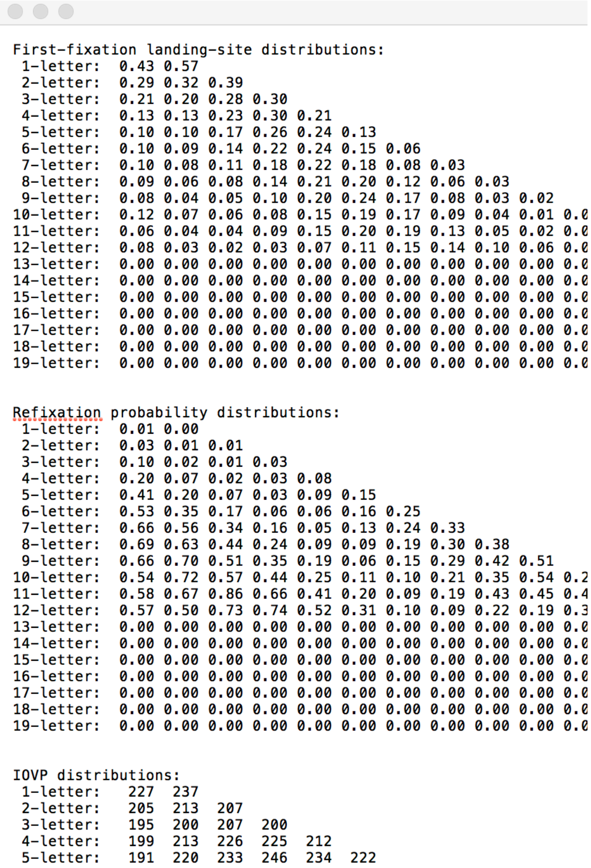
Finally, as Figure 9 shows, the bottom part of the output file contains the mean durations of single fixations as function of their positions (i.e., IOVP curves):



```
0.00 0.00 0.00 0.01  the
0.03 0.03 0.00 0.06  eye

IOVP distributions:
 219  227  247  251  251  235  moved
 186  223  244  238  232  into
 215  247  237  228  her
 199  205  204  202  new
 203  175  211  241  214  240  218  162  185  229  apartment *
 258  274  262  at
 203  190  196  201  the
 209  240  273  267  end
 217  206  174  of
 157  142  135  136  the
   0  209  235  252  250  259  257  240  218  203  principal
 293  269  300  317  316  316  311  311  294  249  154  introduced
 204  204  207  211  the
 241  255  258  230  new
 144  191  217  276  300  297  287  274  277  288  president *
```

Figure 9. Final example of output from “Word DVs” simulation, showing IOVP curves for each word.

(5.) Distributions – As Figure 10 shows, this final type of simulation will generate three distributions across words of each given length: (1) first-fixation landing-site distributions; (2) refixation-probability distributions; and (3) IOVP curves.



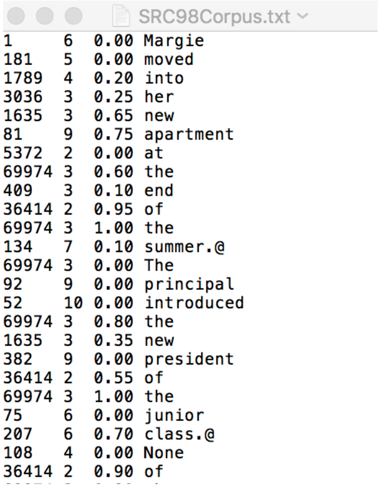
```
First-fixation landing-site distributions:
 1-letter:  0.43 0.57
 2-letter:  0.29 0.32 0.39
 3-letter:  0.21 0.20 0.28 0.30
 4-letter:  0.13 0.13 0.23 0.30 0.21
 5-letter:  0.10 0.10 0.17 0.26 0.24 0.13
 6-letter:  0.10 0.09 0.14 0.22 0.24 0.15 0.06
 7-letter:  0.10 0.08 0.11 0.18 0.22 0.18 0.08 0.03
 8-letter:  0.09 0.06 0.08 0.14 0.21 0.20 0.12 0.06 0.03
 9-letter:  0.08 0.04 0.05 0.10 0.20 0.24 0.17 0.08 0.03 0.02
10-letter:  0.12 0.07 0.06 0.08 0.15 0.19 0.17 0.09 0.04 0.01 0.0
11-letter:  0.06 0.04 0.04 0.09 0.15 0.20 0.19 0.13 0.05 0.02 0.0
12-letter:  0.08 0.03 0.02 0.03 0.07 0.11 0.15 0.14 0.10 0.06 0.0
13-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
14-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
15-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
16-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
17-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
18-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
19-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0


Refixation probability distributions:
 1-letter:  0.01 0.00
 2-letter:  0.03 0.01 0.01
 3-letter:  0.10 0.02 0.01 0.03
 4-letter:  0.20 0.07 0.02 0.03 0.08
 5-letter:  0.41 0.20 0.07 0.03 0.09 0.15
 6-letter:  0.53 0.35 0.17 0.06 0.06 0.16 0.25
 7-letter:  0.66 0.56 0.34 0.16 0.05 0.13 0.24 0.33
 8-letter:  0.69 0.63 0.44 0.24 0.09 0.09 0.19 0.30 0.38
 9-letter:  0.66 0.70 0.51 0.35 0.19 0.06 0.15 0.29 0.42 0.51
10-letter:  0.54 0.72 0.57 0.44 0.25 0.11 0.10 0.21 0.35 0.54 0.2
11-letter:  0.58 0.67 0.86 0.66 0.41 0.20 0.09 0.19 0.43 0.45 0.4
12-letter:  0.57 0.50 0.73 0.74 0.52 0.31 0.10 0.09 0.22 0.19 0.3
13-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
14-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
15-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
16-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
17-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
18-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0
19-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0


IOVP distributions:
 1-letter:   227  237
 2-letter:   205  213  207
 3-letter:   195  200  207  200
 4-letter:   199  213  226  225  212
 5-letter:   191  220  233  246  234  222
```

Figure 10. Example output of “Distributions” simulation.

2. Setting Up Sentence and Target-Word Files

As indicated previously, two files are required to run simulations: (1) a sentence file containing information about each word’s frequency, length, cloze predictability, and identity; and (2) a file identifying a specific target word in each sentence. This section describes how to set up these files to run simulations on sentences other than the Schilling et al. (1998) corpus.

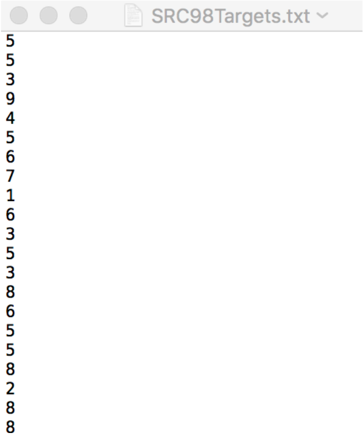
The sentence file should contain four columns of information about each word’s: (1) frequency of occurrence; (2) length in character spaces; (3) cloze predictability; and (4) identity. The last word of each sentence should also be followed by an ampersand (i.e., @), as indicated in Figure 11, below. Without this marker, the program will treat all of the words in the file are a single sentence, which may or may not be useful. (For more information about the Schilling et al., 1998 sentence corpus, see Reichle et al., 1998.)



```
1     6  0.00 Margie
181   5  0.00 moved
1789  4  0.20 into
3036  3  0.25 her
1635  3  0.65 new
81    9  0.75 apartment
5372  2  0.00 at
69974 3  0.60 the
409   3  0.10 end
36414 2  0.95 of
69974 3  1.00 the
134   7  0.10 summer.@
69974 3  0.00 The
92    9  0.00 principal
52    10 0.00 introduced
69974 3  0.80 the
1635  3  0.35 new
382   9  0.00 president
36414 2  0.55 of
69974 3  1.00 the
75    6  0.00 junior
207   6  0.70 class.@
108   4  0.00 None
36414 2  0.90 of
```

Figure 11. Example of sentence file.

The target-word file is just a list identifying target words, as shown in Figure 12. The file contains a single column containing one number per sentence. (Following Java conventions, the numbers range from 0 to *N*-1 for a sentence containing *N* words; e.g., the "5" in the first row specifies the sixth word as the target for the first sentence.) These target words will be tagged in the simulation output (with asterisks) to make analyses of those words easier. However, if you are not interest in specific target words, this file can be set up with "dummy" numbers (e.g., a single column of 0s).



```
5
5
3
9
4
5
6
7
1
6
3
5
3
8
6
5
5
8
2
8
8
```

Figure 12. Example of target-word file.

The model program should be fairly robust and handle slight variations in formatting (e.g., using blank spaces vs. tabs between columns). However, it's a good idea to make sure that the model is reading in the files correctly using the *Word IVs* output option before you run any real simulations. Also, the sentence and target-word files should be an ascii file (i.e., a file that only contains alphanumeric characters, with no hidden control characters.) Finally, it’s important to remember that fixations on the first and last word of each sentence are excluded from analyses because the processing of these words starts and ends (respectively) abruptly.

Finally, although the code has been extensively tested to ensure that there are no bugs, it’s important for you to understand both how the program works and what the model does and doesn’t do. For example, as indicated in Reichle, Warren, and McConnell (2009), when integration fails for a particular word, the model moves its attention and eyes back to the word so it can be identified and integrated a second time. The assumption here is that this second attempt at lexical processing and integration requires (on average) the same length of time to complete as the first attempt—an assumption that is made for convenience and that is unlikely to be valid. This assumption therefore also means that any of the model’s predictions about second-pass reading-time measures (e.g., go-past times) have to be treated with some skepticism. To make this as explicit as I can: The model is acknowledged to have limitations, with one of these being that the model’s integration stage “in a placeholder for a deeper theory of post-lexical language processing during reading” (p. 7). I make this disclaimer again here because, although I’ve done my absolute best to make the model both conceptually transparent and easy to use (e.g., as evidenced by my providing both the source code and a GUI version of the model), it’s impossible to anticipate all of the ways in which the model might be used, so the responsibility for using it in a manner that respects the model’s assumptions is yours. With that important caveat in mind, I sincerely hope that you will use the model for its intended purposes: running simulations to facilitate the interpretation of eye-tracking data, to test hypotheses, and—if you’re lucky!—to falsify the model’s assumptions and thereby advance the field of reading research.

Don’t hesitate to contact me if you have any questions or run into any problems. Good luck!

Best regards,
Erik