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 me from my website (www.erikdreichle.com) and upon request (reichle@soton.ac.uk). 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 actual model (*E-Z Reader 10.2.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.

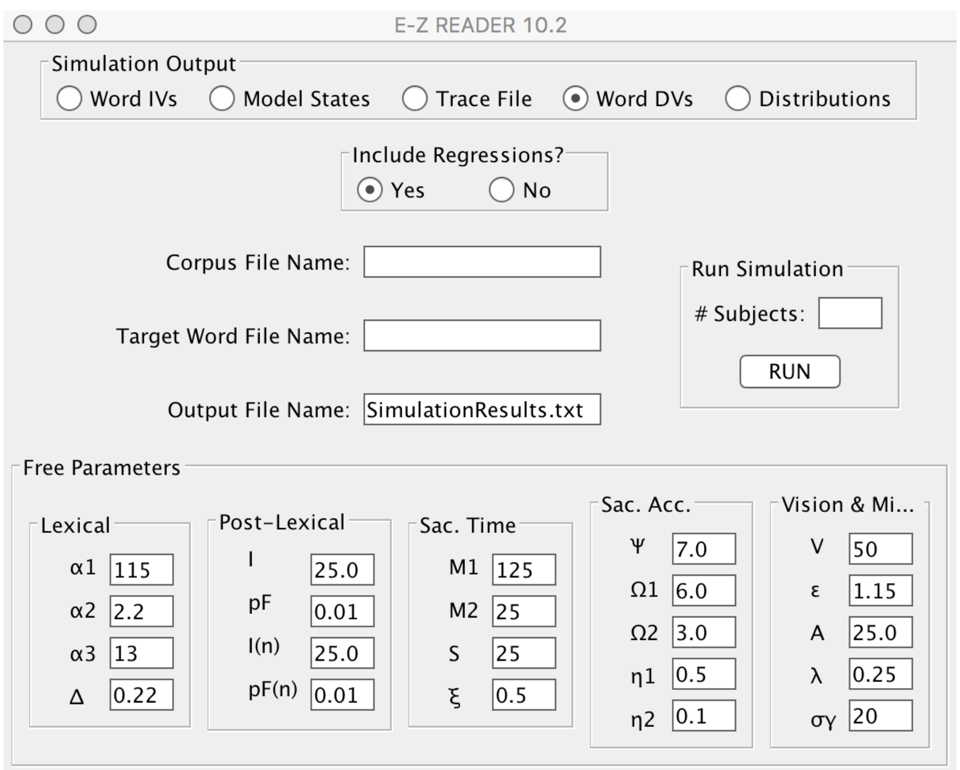


Figure 1. E-Z Reader GUI.

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 by me and my colleagues to evaluate different versions of the E-Z Reader model.

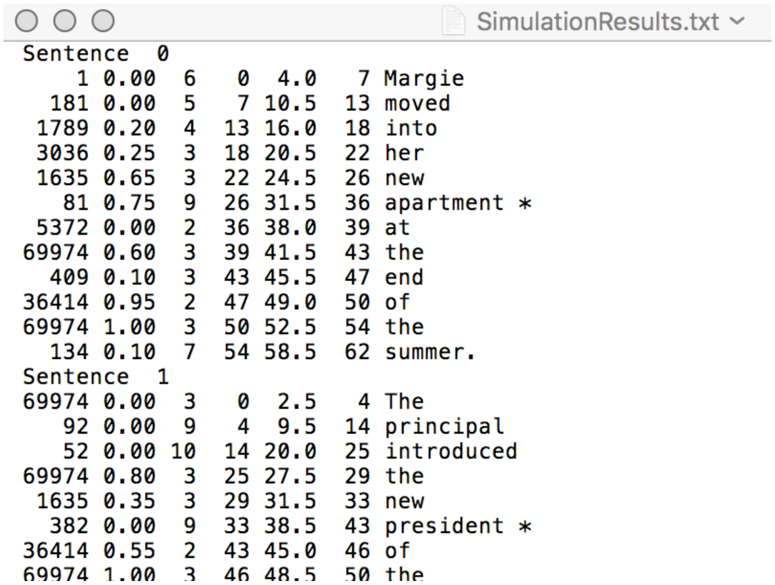
(2.) *# Subjects* – Enter the number of subjects (1-10,000) that will be used in completing the simulation.

(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).

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. 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 2 shows an example of the first part of the output from this type of simulation and an explanation of what it means:


SimulationResults.txt

```
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 *
36414 0.55  2  43 45.0  46 of
69974 1.00  3  46 48.5  50 the
```

Figure 2. 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 program to write out all of the internal states that the model 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 3 shows an example of the output and an explanation of what it means:

```
S:  0 N:  0 FixN:  0 FixW:  0 FixPos:  4.0 FixDur:    0 Rate: 1.32 - L1 183
S:  0 N:  0 FixN:  0 FixW:  0 FixPos:  4.0 FixDur:  183 Rate: 1.32 - L2  19 M1 121 (1 6.5)
S:  0 N:  0 FixN:  0 FixW:  0 FixPos:  4.0 FixDur:  202 Rate: 1.32 -  I  26  A  20 M1 101 (1 6.5)
S:  0 N:  1 FixN:  0 FixW:  0 FixPos:  4.0 FixDur:  222 Rate: 2.97 - L1 250  I   6 M1  82 (1 6.5)
S:  0 N:  1 FixN:  0 FixW:  0 FixPos:  4.0 FixDur:  228 Rate: 2.97 - L1 244 M1  75 (1 6.5)
S:  0 N:  1 FixN:  0 FixW:  0 FixPos:  4.0 FixDur:  304 Rate: 2.97 - L1 169 M2  22 (5.6)
S:  0 N:  1 FixN:  0 FixW:  0 FixPos:  4.0 FixDur:  325 Rate: 2.97 - L1 147  S  25
S:  0 N:  1 FixN:  1 FixW:  1 FixPos:  9.6 FixDur:    0 Rate: 2.97 -  V  50 L1 122
S:  0 N:  1 FixN:  1 FixW:  1 FixPos:  9.6 FixDur:   50 Rate: 1.32 - L1  32
S:  0 N:  1 FixN:  1 FixW:  1 FixPos:  9.6 FixDur:   82 Rate: 1.32 - L2  22 M1 160 (2 6.4)
S:  0 N:  1 FixN:  1 FixW:  1 FixPos:  9.6 FixDur:  104 Rate: 1.32 -  I  25  A  20 M1 137 (2 6.4)
S:  0 N:  2 FixN:  1 FixW:  1 FixPos:  9.6 FixDur:  124 Rate: 3.08 - L1 362  I   5 M1 117 (2 6.4)
S:  0 N:  2 FixN:  1 FixW:  1 FixPos:  9.6 FixDur:  129 Rate: 3.08 - L1 357 M1 112 (2 6.4)
S:  0 N:  2 FixN:  1 FixW:  1 FixPos:  9.6 FixDur:  242 Rate: 3.08 - L1 245 M2  30 (7.1)
S:  0 N:  2 FixN:  1 FixW:  1 FixPos:  9.6 FixDur:  271 Rate: 3.08 - L1 215  S  25
S:  0 N:  2 FixN:  2 FixW:  2 FixPos: 16.7 FixDur:    0 Rate: 3.08 -  V  50 L1 190
S:  0 N:  2 FixN:  2 FixW:  2 FixPos: 16.7 FixDur:   50 Rate: 1.26 - L1  58
S:  0 N:  2 FixN:  2 FixW:  2 FixPos: 16.7 FixDur:  108 Rate: 1.26 - L2  25 M1  96 (3 3.8)
S:  0 N:  2 FixN:  2 FixW:  2 FixPos: 16.7 FixDur:  133 Rate: 1.26 -  I  27  A  26 M1  70 (3 3.8)
S:  0 N:  3 FixN:  2 FixW:  2 FixPos: 16.7 FixDur:  159 Rate: 2.02 - L1 259  I   1 M1  44 (3 3.8)
S:  0 N:  2 FixN:  2 FixW:  2 FixPos: 16.7 FixDur:  160 Rate: 2.02 -  A  30 M1  95 (2 -0.7)
S:  0 N:  3 FixN:  2 FixW:  2 FixPos: 16.7 FixDur:  190 Rate: 2.02 - L1 238 M1  65 (2 -0.7)
S:  0 N:  3 FixN:  2 FixW:  2 FixPos: 16.7 FixDur:  255 Rate: 2.02 - L1 173 M2  37 (-2.1)
```

Figure 3. Example output from "Model States" simulation.

The above example shows consecutive model states, displayed one per row. Within each row going from left to right are: (1) the current sentence being read (e.g., the first sentence, or "S: 0", in this example); (2) an index "N" of where attention is located (i.e., the word being processed); (3) "FixN", the fixation number; (4) "FixW", the word currently being fixated; (5) "FixPos", the cumulative character position of the current fixation location; (6) "FixDur", the duration of the current fixation; (7) "Rate", the current rate of lexical processing; and (8) a list of on-going processes and their associated completion times (in ms). These on-going processes are: (1) "V", pre-attentive vision; (2) "L1", the first stage of lexical processing (i.e., the familiarity check); (3) "L2", the second stage of lexical processing (i.e., lexical access); (4) "I", post-lexical integration; (5) "A", attention shift; (6) "M1", labile saccadic programming; (7) "M2", non-labile saccadic programming; and (8) "S", the saccade. For "M1", the two numbers in parentheses respectively indicate the word being targeted by the saccade and its intended length. For "M2", the number in parentheses indicates the saccade length after both random and systematic error have been added to its intended length. For example, in the second line, the duration of M1 is 121 ms, the impending saccade will be being directed towards the center of word 1 (i.e., its OVP), with an intended length of 6.5 character spaces. However, as line 6 shows, the actual saccade length is only 5.6 character spaces, which as line 8 then shows, moves the eyes from the OVP of word 0 (i.e., cumulative character position 4.0) to cumulative character position 9.6 in word 1. For a detailed discussion of the model states and how state transitions occur in E-Z Reader, see Reichle, Pollatsek, Fisher, and Rayner (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 in experiments involving human participants. Figure 4 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.

SimulationRe

```
dur: 321 pos:  4.0 word:  0  Margie
dur: 249 pos: 11.6 word:  1  moved
dur: 247 pos: 14.7 word:  2  into
dur: 295 pos: 22.3 word:  4  new
dur: 154 pos: 35.1 word:  5  apartment
dur: 195 pos: 38.9 word:  6  at
dur: 248 pos: 46.8 word:  8  end
dur: 153 pos: 55.5 word: 11  summer.
dur: 263 pos:  2.5 word:  0  The
dur: 265 pos:  9.5 word:  1  principal
dur: 352 pos: 19.2 word:  2  introduced
dur: 288 pos: 30.5 word:  4  new
dur: 257 pos: 38.0 word:  5  president
dur: 130 pos: 46.0 word:  6  of
dur: 266 pos: 50.3 word:  8  junior
dur: 289 pos:  3.0 word:  0  None
dur: 203 pos: 14.3 word:  3  students
dur:  88 pos: 19.0 word:  3  students
dur: 127 pos: 29.7 word:  5  to
dur: 222 pos: 24.9 word:  4  wanted
dur: 309 pos: 30.4 word:  5  to
dur: 227 pos: 40.2 word:  8  exam
dur: 147 pos: 43.6 word:  8  exam
dur: 242 pos: 49.0 word:  9  after
dur: 237 pos:  3.0 word:  0  Mark
```

Figure 4. 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 lexical access 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, Pollatsek, Fisher & Rayner, 1998). Figures 5-8 provide examples of the simulation output and an explanation of what it means:

SimulationResults.txt

```
Word-based means:
SFD 248 FFD 242 GD 256 TT 265 PrF 0.97 Pr1 0.91 Pr2 0.06 PrS 0.03 moved
SFD 230 FFD 228 GD 237 TT 250 PrF 0.77 Pr1 0.74 Pr2 0.03 PrS 0.23 into
SFD 231 FFD 230 GD 238 TT 251 PrF 0.63 Pr1 0.60 Pr2 0.02 PrS 0.37 her
SFD 213 FFD 212 GD 224 TT 240 PrF 0.40 Pr1 0.38 Pr2 0.02 PrS 0.60 new
SFD 215 FFD 205 GD 260 TT 276 PrF 0.68 Pr1 0.53 Pr2 0.15 PrS 0.32 apartment *
SFD 267 FFD 266 GD 268 TT 294 PrF 0.57 Pr1 0.57 Pr2 0.01 PrS 0.43 at
SFD 202 FFD 205 GD 218 TT 247 PrF 0.59 Pr1 0.54 Pr2 0.04 PrS 0.41 the
SFD 263 FFD 259 GD 267 TT 283 PrF 0.68 Pr1 0.65 Pr2 0.03 PrS 0.32 end
SFD 201 FFD 202 GD 204 TT 207 PrF 0.24 Pr1 0.23 Pr2 0.00 PrS 0.77 of
SFD 129 FFD 129 GD 129 TT 130 PrF 0.13 Pr1 0.13 Pr2 0.00 PrS 0.87 the
SFD 254 FFD 242 GD 287 TT 287 PrF 1.00 Pr1 0.88 Pr2 0.12 PrS 0.00 principal
SFD 312 FFD 267 GD 324 TT 325 PrF 0.95 Pr1 0.68 Pr2 0.28 PrS 0.05 introduced
SFD 213 FFD 213 GD 217 TT 242 PrF 0.25 Pr1 0.25 Pr2 0.01 PrS 0.75 the
```

Figure 5. 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).

As Figure 6 shows, the next part of the output file contains the first-fixation landing-site distributions for each word:

SimulationResults.txt

```
First-fixation landing-site distributions:
0.01 0.09 0.20 0.35 0.26 0.09  moved
0.07 0.08 0.26 0.39 0.20  into
0.13 0.19 0.34 0.34  her
0.36 0.18 0.23 0.23  new
0.13 0.05 0.06 0.08 0.13 0.10 0.10 0.11 0.11 0.14  apartment *
0.27 0.35 0.39  at
0.35 0.27 0.20 0.18  the
0.17 0.24 0.30 0.29  end
0.55 0.30 0.15  of
0.18 0.25 0.28 0.29  the
0.00 0.00 0.02 0.10 0.23 0.34 0.21 0.08 0.02 0.00  principal
0.01 0.02 0.05 0.12 0.21 0.24 0.21 0.09 0.04 0.00 0.00  introduced
0.12 0.17 0.32 0.40  the
```

Figure 6. Second example of output from "Word DVs" simulation, showing first-fixation landing-site distributions for each word.

As Figure 7 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:

SimulationResults.txt

```
Refixation probability distributions:
0.08 0.18 0.04 0.01 0.10 0.14  moved
0.19 0.05 0.02 0.01 0.06  into
0.11 0.06 0.00 0.03  her
0.11 0.03 0.00 0.07  new
0.44 0.66 0.68 0.49 0.21 0.06 0.01 0.12 0.01 0.03  apartment *
0.03 0.00 0.01  at
0.16 0.04 0.01 0.03  the
0.16 0.02 0.01 0.04  end
0.02 0.00 0.00  of
0.00 0.00 0.00 0.00  the
0.33 0.00 0.38 0.20 0.12 0.04 0.11 0.25 0.41 0.25  principal
1.00 0.91 0.78 0.56 0.32 0.10 0.07 0.29 0.44 0.00 0.50  introduced
0.07 0.02 0.01 0.01  the
```

Figure 7. Third example of output from "Word DVs" simulation, showing refixation-probability distributions for each word.

Finally, as Figure 8 shows, the bottom part of the output file contains the mean durations of single fixations as function of their positions (i.e., IOVP curves):

SimulationResults.txt

```
IOVP distributions:
 196  232  245  256  252  moved
 179  217  239  234  into
 215  226  239  her
 215  213  213  new
 201  191  237  267  256  228  212  179  171  apartment *
 263  279  at
 212  198  193  the
 230  263  278  end
 213  200  of
 152  124  130  the
 164  193  241  244  258  259  251  251  255  principal
   0  289  295  316  320  312  312  313  290  191  introduced
 191  211  228  the
```

Figure 8. Final example of output from "Word DVs" simulation, showing IOVP curves for each word.

(5.) Distributions – As Figure 9 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.

SimulationResults.txt

```
First-fixation landing-site distributions:
 1-letter:  0.44 0.56
 2-letter:  0.29 0.32 0.39
 3-letter:  0.22 0.21 0.28 0.30
 4-letter:  0.13 0.13 0.23 0.30 0.20
 5-letter:  0.10 0.10 0.18 0.26 0.24 0.12
 6-letter:  0.10 0.09 0.14 0.22 0.24 0.15 0.06
 7-letter:  0.10 0.09 0.12 0.18 0.23 0.17 0.08 0.03
 8-letter:  0.09 0.06 0.08 0.14 0.20 0.20 0.13 0.05 0.03
 9-letter:  0.07 0.05 0.05 0.10 0.20 0.24 0.18 0.07 0.03 0.01
10-letter:  0.11 0.07 0.06 0.09 0.15 0.20 0.18 0.09 0.04 0.01 0.01
11-letter:  0.07 0.03 0.05 0.09 0.15 0.20 0.19 0.13 0.05 0.02 0.01 0.01
12-letter:  0.08 0.03 0.02 0.03 0.06 0.12 0.14 0.15 0.11 0.07 0.06 0.07
13-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
14-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
15-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

Refixation probability distributions:
 1-letter:  0.01 0.00
 2-letter:  0.03 0.01 0.01
 3-letter:  0.09 0.02 0.01 0.03
 4-letter:  0.19 0.06 0.02 0.03 0.07
 5-letter:  0.36 0.19 0.06 0.02 0.09 0.16
 6-letter:  0.48 0.33 0.16 0.05 0.06 0.16 0.25
 7-letter:  0.65 0.53 0.33 0.15 0.04 0.13 0.24 0.33
 8-letter:  0.61 0.54 0.44 0.25 0.08 0.08 0.20 0.31 0.39
 9-letter:  0.57 0.67 0.53 0.36 0.18 0.06 0.15 0.27 0.42 0.45
10-letter:  0.59 0.52 0.56 0.41 0.23 0.10 0.08 0.19 0.33 0.44 0.13
11-letter:  0.56 0.64 0.71 0.65 0.43 0.20 0.06 0.17 0.35 0.61 0.52 0.44
12-letter:  0.72 0.76 0.65 0.71 0.55 0.30 0.14 0.10 0.18 0.24 0.33 0.30
13-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
14-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
15-letter:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

IOVP distributions:
 1-letter:   227  237
 2-letter:   205  216  210
 3-letter:   195  202  207  198
 4-letter:   199  216  227  225  213
 5-letter:   197  221  236  247  233  222
 6-letter:   191  228  247  257  257  241  227
 7-letter:   179  240  265  274  276  264  248  228
 8-letter:   192  231  271  272  277  272  259  244  221
```

Figure 9. 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 10, 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.)

SRC98Corpus.txt

```
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.@
```

Figure 10. Example of sentence file.

The target-word file is just a list identifying target words, as shown in Figure 11. 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).

SRC98Targets.txt

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

Figure 11. 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 *WordIVs* 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, and 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 lexical processing of these words starts and ends (respectively) abruptly.

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

Best regards,
Erik