Running E-Z Reader 10.3 Simulations

The E-Z Reader 10.3 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 actual 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, or 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.

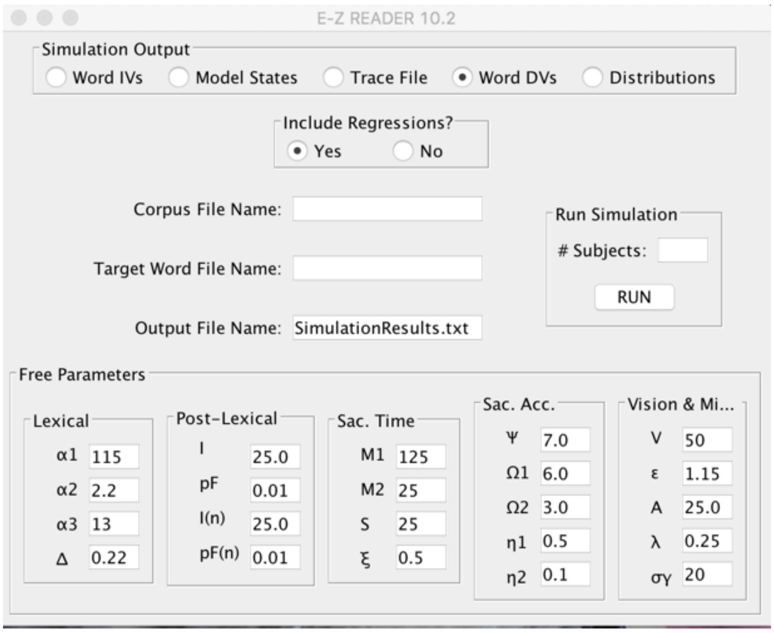

Figure 1. E-Z Reader GUI.

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 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) *Target File Name* – Enter the name of the file containing the pre-specified target words in each sentence. For example, Schilling et al.'s (1998) experiment examined the effects of word frequency by examining how low- and high-frequency target affected various eye-movement measures; these specific target words are listed (one per sentence) in *SRC98Targets.txt*.

(3) *Output File Name* – This field designates the file where simulation results will be written (default name: *SimulationResults.txt*). This file will appear in the same location as the program file after the program executes. (Note that changing the *.txt* file extension to *.xls* will cause the output to be formatted as a space delimited Microsoft EXCEL file, making the output easier to analyze.)

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

(5) *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, including the number of statistical subjects, number of sentences, type of simulation, etc.

(6) *Include Regressions?* – This button determines whether (yes) or not (no) simulation trials containing inter-word regressions will be included in the output.

(7) *Parameter Values* – These fields 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 respectfully used to set the values of "I" and "pF" for the specific target words. Second, the parameter that controls the gamma distribution variability, "σγ", is set equal to a value (=20) that generates gamma distributions with 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.)

(8) *Simulation Output* – These buttons can be selected to execute a variety of different simulations, each providing a different type of output:

(a) *Word IVs* – Selecting this button will output the independent variables associated with each of the words in the sentence file. Because these variables are calculated by the program prior to executing a simulation (e.g., each word's *optimal viewing position*, or *OVP*), it's good practice 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 output that is generated:

```
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
   75 0.00  6   50 54.0   57 junior
  207 0.70  6   57 61.0   64 class.
Sentence  2
```

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: (i) its frequency of occurrence in printed text (e.g., as tabulated by Francis & Kucera, 1982); (ii) its cloze predictability (Taylor, 1953); (iii) its length (i.e., number of letters); (iv) the cumulative character position of the space immediately to the left of the word; (v) the cumulative character position of the center of the word (i.e., its OVP); (vi) the cumulative character position of the right side the last character in a word; (vii) the actual word, with asterisks marking target words.

(b) *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 for figuring out precisely 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 generated:

```
Simula
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:
```

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 the following are indicated: (i) the current sentence being read (e.g., the first sentence, or "S: 0", in this example); (ii) an index "N" of where attention is located (i.e., the word being processed); (iii) "fix#", the fixation number; (iv) "word", the word currently being fixated; (v) "pos", the cumulative character position of the current fixation location; (vi) "dur", the duration of the current fixation; (vii) "pr", the current lexical processing rate; (viii) a list of on-going processes and their associated completion times (in ms), with the first process listed in square brackets (e.g., [L1] in line 2) being the process that has just completed; and (ix) an indicator, "IF:", of those words (if any) for which integration failure may have occurred.

These on-going processes 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.

For "M1", the two numbers in parentheses respectively indicate the word being targeted by the saccade and its intended length. Similarly, 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 81 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 4.9 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 8.9 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).

(c) *Trace* – Selecting this button will cause the model to generate 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: (i) its duration (*dur*); (ii) its position (*pos*); (iii) the word being fixated (*word*); and (iv) the word.

SimulationResults.txt

```
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
dur: 138 pos: 47.4 word:  7  the
dur: 182 pos: 41.3 word:  5  president
dur: 309 pos: 53.2 word:  8  junior
dur: 331 pos:  3.0 word:  0  None
dur: 158 pos: 15.5 word:  3  students
dur: 211 pos: 19.4 word:  3  students
dur: 303 pos: 25.4 word:  4  wanted
```

Figure 4. Example output of "Trace File" simulation.

(d) *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 the last word has been integrated. (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:

SimulationResults.txt

```
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
```

Figure 5. First example 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: (i) single-fixation duration (SFD); (ii) first-fixation duration (FFD); (iii) gaze duration (GD); (iv) total time (TT); (v) fixation probability (PrF); (vi) probability of making exactly one fixation (Pr1); (vii) probability of making two or more fixations (Pr2); (viii) probability of skipping (PrS); and (ix) the word, with target words indicated by asterisks.

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

```
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
```

Figure 6. Second example output from "Word DVs" simulation, showing the 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 as a function of initial fixation position) for each word:

```
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
```

Figure 7. Third example 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):

```
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
```

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

(e) *Distributions* – As Figure 9 shows, this final type of simulation will generate three distributions across words of each given length: (i) first-fixation landing-site distributions; (ii) refixation-probability distributions; and (iii) 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
 6-letter:   194  229  246  259  256  241  226
 7-letter:   180  234  266  274  276  266  249  230
 8-letter:   183  214  267  273  274  273  258  242  212
 9-letter:   166  184  252  278  284  288  275  262  243  184
```

Figure 9. Example output from “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 using 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
```

Figure 10. Example of sentence file.

The target-word file is 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 their analyses 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 ascii files (i.e., files 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 the 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 problems. Good luck!

Best regards,
Erik

References

Francis, W. N. & Kucera, H. (1982). *Frequency analysis of English usage: Lexicon and grammar*. Boston: Houghton Mifflin.

Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. (1992). *Numerical recipes in C: The art of scientific computing*. New York: Cambridge University Press.

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

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 & Cognition, 26*, 1270-1281.

Taylor, W. L. (1953). Cloze procedure: A new tool for measuring readability. *Journalism Quarterly*, *30*, 415-433.