Running E-Z Reader 9 Simulations

The program for running simulations using the E-Z Reader model was written in Java, version 1.5. The executable (.jar) version of the program is available at: www.pitt.edu/~reichle/ezreader.html. The source code (i.e., .java classes) is also available from me (Erik Reichle, at: reichle@pitt.edu) upon request. The first part of these instructions describes how to use the program file to run simulations. The second part describes how to set up your own sentence/text files.

1. Running Simulations

You will need two files to run E-Z Reader simulations: (1) the program file containing the actual model (*E-Z_Reader_9.jar*), and (2) a file containing the sentences/text that will be used in the simulation (e.g., *SRC98Corpus.txt*). To run a simulation, first download both files (or provide your own sentence/text file) to a common location on your computer (e.g., its desktop or common folder) and then double-click on the program file. This should open a graphic-user interface (GUI) 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, the different types of simulations that can be run, and of how to create your own sentence/text files:

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/text that will be used in the simulation. The file that is provided on my website (*SRC98Corpus.txt*) contains the sentences that were first used by Schilling, Rayner, and Chumbley (1998) in their eye-movement experiment, and that have subsequently been used by me and my colleagues to evaluate different versions of the E-Z Reader model.

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

(3.) *RUN* – Press this button to start the simulation. The progress bar at the bottom of the GUI will indicate how much of the simulation has completed. The length of time that is needed to complete a simulation will obviously depend upon the speed of your computer, and on both the number of statistical subjects and the length of sentences/text being used in the simulation. With my 2.33 GHz MacBook Pro, a simulation using the Schilling et al. (1998) sentences and 1,000 statistical subjects takes approximately 9 seconds to complete.

All of the other buttons and text fields are set to default values, but can be modified as necessary. The function of the text field labeled *Output File Name* is fairly obvious; the results of any simulation are written to this file (default name: *Simulation Results.txt*), which will appear in the same location as the program and sentence/text files. (Note that changing the *.txt* file extension to *.xls* will cause the output to be automatically written to a space-delimited Microsoft EXCEL file that—in most cases—will make the output much easier to analyze.) The functions of the text fields in the box labeled *Parameter Values* is also pretty obvious; you can change the default parameter values of the E-Z Reader model by entering new values. For a description of the parameters and their interpretation, see Pollatsek, Reichle, and Rayner (2006) or Reichle, Rayner, and Pollatsek (2006). (Note that the parameter that controls the variability of the gamma distributions, $\sigma_\gamma = 20$, is set equal to a value that generates gamma distributions having standard deviations that are 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?* allow different types of simulation results to be written to the output file after a simulation has completed. The different types of output are as follows:

(1.) *Word IVs* – Selecting this button will cause all of the independent variables that are contained in the sentence/text file and those that are calculated by the model program prior to executing a simulation (e.g., each word's optimal viewing position) to be written to the output file. It's a good idea to run a simulation with this button selected prior to completing any other simulations to ensure that the sentence/text file has been formatted correctly. (It is also a good idea to use a very small number of statistical subjects to avoid creating a really big text file.) Here is an example of the simulation output and an explanation of what it means:

Sentence: 0

#	Word	Freq	lnFreq	Class	Pred	Length	Char1	CharN	OVP
0	Margie	1	0.00	1	0.0	7.0	0.0	6.0	3.5
1	moved	181	5.20	3	0.0	6.0	7.0	12.0	10.0
2	into	1789	7.49	4	0.2	5.0	13.0	17.0	15.5
3	her	3036	8.02	4	0.25	4.0	18.0	21.0	20.0
4	new	1635	7.40	4	0.65	4.0	22.0	25.0	24.0
5	apartment	81	4.39	2	0.75	10.0	26.0	35.0	31.0
6	at	5372	8.59	4	0.0	3.0	36.0	38.0	37.5
7	the	69974	11.16	5	0.6	4.0	39.0	42.0	41.0
8	end	409	6.01	3	0.1	4.0	43.0	46.0	45.0
9	of	36414	10.50	5	0.95	3.0	47.0	49.0	48.5
10	the	69974	11.16	5	1.0	4.0	50.0	53.0	52.0
11	summer.	134	4.90	3	0.1	8.0	54.0	61.0	58.0

Sentence: 1

#	Word	Freq	lnFreq	Class	Pred	Length	Char1	CharN	OVP
0	The	69974	11.16	5	0.0	4.0	0.0	3.0	2.0
1	principal	92	4.52	2	0.0	10.0	4.0	13.0	9.0

.
.
.

The above example shows the first sentence (Sentence: 0) and part of the second (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. The columns from left to right are: (1) word number (#); (2) the actual word (Word); (3) its frequency (Freq); (4) the natural logarithm of its frequency (lnFreq); (5) its frequency class (Class); (6) its predictability (Pred); (7) its length (Length); the cumulative character position of (8) the first character of the word (Char0); (9) the last character (CharN) of the word; and (10) the word's optimal viewing position (OVP). A few notes about the latter variables: First, as per convention, word length includes the blank character space to the left of each word. Second, characters are centered on the indicated character position numbers. For example, the first "a" in the word "apartment" is located at cumulative character position 26, which is the 27th character position from the blank space to the left of "Margie" (which is character position 0). This means that any fixation that happens to land on x, where $25.5 \leq x < 26.5$, will be scored as having been a fixation on character position 26.

(2.) *Model States* – Selecting this button will cause the model program to write out all of the internal states that transpire as the model progresses through the sentences/text. 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 the predictions that it does. Because the output files are very large (each word that is processed might cause the model to go through 6-10 states), it is a good idea to use only a very small number of subjects (e.g., *N* = 1) when running simulations of this type. Here is an example of the simulation output and an explanation of what it means:

```
 Sentence: 0
 Active Processes:
     L1:  duration: 245.37 ms; processing rate: 1.34
 Attention: 0
 Current Fixation:
     word #: 0
     cumulative character #: 3.50
     duration: 0.00 ms
-----------------------------
 Sentence: 0
 Active Processes:
     L2:  duration: 62.50 ms
     M1:  duration: 93.13 ms (engage time: 46.57 ms; convert time: 46.57 ms); word target #: 1
 Attention: 0
 Current Fixation:
     word #: 0
     cumulative character #: 3.50
     duration: 245.37 ms
-----------------------------
 Sentence: 0
 Active Processes:
     M1:  duration: 30.63 ms (engage time: 0.00 ms; convert time: 30.63 ms); word target #: 1
     L1:  duration: 223.65 ms; processing rate: 2.74
 Attention: 1
 Current Fixation:
     word #: 0
     cumulative character #: 3.50
     duration: 307.87 ms
-----------------------------
 Sentence: 0
```

Active Processes:
L1: duration: 193.02 ms; processing rate: 2.74
M2: duration: 22.40 ms; target word #: 1; intended saccade length: 6.50 characters
Attention: 1
Current Fixation:
word #: 0
cumulative character #: 3.50
duration: 338.50 ms

.
.
.

The above example shows four consecutive model states, each separated by a row of dashes. At the top of each state is the current sentence (the first sentence, or "Sentence: 0", in this example). Below that is a list of all active processes. For V, L_2, V, and R, only the durations (in ms) are shown. For L_1, both the duration and the rate of processing (which varies as a function of retinal eccentricity) are shown. For M_1, three different times are shown: the overall process duration, the time required to engage the oculomotor system (i.e., engage time), the time to convert the (word) spatial target into a distance metric (i.e., convert time). The actual saccade target (i.e., the target word's number) is also shown for M_1. Similarly, for M_2, the duration, the saccade target, and the intended saccade length (in character spaces) are shown. The word being attended is shown right below the active processes. Finally, the bottom three lines show the current fixation location (both in terms of the word number being fixated and the cumulative character position) and its duration (in ms). For a detailed discussion of the model states and how state transitions occur in (an earlier version of) E-Z Reader, see Reichle, Pollatsek, Fisher, and Rayner (1998).

(3.) *Fixations* – Selecting this button will result in the output file containing a list of all of the fixations generated by the model—their location in terms of both cumulative character position (i.e., number of characters from the beginning of the sentence) and word number, and their duration (in ms). Again, use a small number of subjects when running this type of simulation. Here is an example of the simulation output and an explanation of what it means:

Sentence: 0; Subject: 0

FixDur	FixLoc	Word#	Fix#
270	4	0	1
211	11	1	2
289	20	3	3
264	29	5	4
224	38	6	5
275	47	9	6

Sentence: 1; Subject: 0

FixDur	FixLoc	Word#	Fix#
282	2	0	1
216	7	1	2
375	17	2	3
310	26	3	4

363	35	5	5
233	44	6	6

Sentence: 2; Subject: 0

FixDur	FixLoc	Word#	Fix#
320	3	0	1
269	10	2	2
421	20	3	3
243	30	5	4
202	36	7	5
273	46	9	6
195	50	10	7

The above output shows the fixations that were generated across three sentences by one statistical subject. The sentence and subject numbers are shown for each sentence. The other information corresponds to individual fixations and is organized into four columns: (1) the duration (in ms) of each fixation (FixDur); (2) its cumulative character location (FixLoc); (3) the word being fixated (Word#); and (4) the fixation number (Fix#).

(4.) *Trace File* – 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 real human participants. This "trace file" will contain the following information:

Sentence: 13; Subject: 0

Word#	#Fix	#1st	FFD	SFD	GD	TT
0	1	1	316	316	316	316
1	2	1	143	0	143	306
2	0	0	0	0	0	0
3	2	1	92	0	92	347
4	1	1	248	248	248	248
5	0	0	0	0	0	0
6	1	1	182	182	182	182
7	1	1	279	279	279	279
8	1	1	335	335	335	335
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0

The above trace file was generated by allowing the inclusion of inter-word regressions, and by increasing the value of λ from 0.09 to 0.25 to encourage such regressions. The top row indicates the subject and subject numbers. The remaining information is organized into seven columns: (1) a given word's number (Word#); (2) total number of fixations on the word (#Fix); (3) the number of first-pass fixations on the word (#1st); (4) the word's first-fixation duration (FFD); (5) single-fixation duration (SFD); (6) gaze duration (GD); and (7) total viewing time (TT).

(5.) *Word DVs* – This is the type of simulation that will probably be most useful for anyone interested in running simulations. Selecting this button will generate an output file that contains a variety of the standard dependent measures (e.g., mean first-fixation durations, gaze durations, etc.) for each word in the sentence/text file. With this type of simulation, it is advisable to use a large number of subjects (e.g., at least 100) to obtain stable simulation results. Also, please remember that the predicted results for the first and last words in each sentence do not accurately reflect the model's performance because the model always starts from the middle of the first word (with no parafoveal preview), and because the model always halts (regardless of whatever is happening) when the second stage of lexical processing (i.e., L_2) on the last word has been completed. (For these reasons, the dependent values of the first and last words are never included in our analyses; see Reichle, Pollatsek, Fisher & Rayner, 1998). Here is an example of the simulation output and an explanation of what it means:

Sentence	FFD	SFD	GD	TT	Pr1	Pr2	Pr3+	PrS	Landing-Site Distributions									
The	265	265	265	265	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00						
hurricane	255	256	276	276	0.95	0.05	0.00	0.00	0.00	0.01	0.05	0.12	0.21	0.25	0.20	0.10	0.04	0.01
destroyed	288	294	337	337	0.79	0.16	0.00	0.05	0.06	0.11	0.15	0.16	0.15	0.16	0.11	0.05	0.04	0.02
houses	263	262	288	288	0.82	0.09	0.00	0.10	0.13	0.16	0.18	0.17	0.17	0.12	0.06			
in	227	227	227	227	0.47	0.00	0.00	0.53	0.26	0.32	0.42							
the	218	216	220	222	0.53	0.01	0.00	0.47	0.38	0.24	0.19	0.19						
village	232	240	274	276	0.77	0.19	0.00	0.04	0.14	0.18	0.18	0.13	0.16	0.11	0.07	0.03		
and	219	217	221	223	0.51	0.01	0.00	0.49	0.29	0.23	0.21	0.27						
left	221	220	225	226	0.73	0.02	0.00	0.26	0.23	0.20	0.19	0.20	0.17					
many	198	197	202	202	0.60	0.01	0.00	0.39	0.26	0.17	0.20	0.22	0.15					
homeless.	173	176	180	180	0.32	0.02	0.00	0.66	0.35	0.23	0.22	0.14	0.05	0.00	0.00	0.00	0.01	0.00

The top row lists the dependent variables (means) that are displayed to the right of each word: (1) first-fixation duration (FFD); (2) single-fixation duration (SFD); (3) gaze duration (GD); (4) total time (TT); (5) probability of making one fixation (Pr1); (6) probability of making exactly two fixations (Pr2); (7) probability of making three or more fixations (Pr3+); (8) probability of skipping (PrS); and (9) the mean proportion of first fixations landing on each of the word's character positions (i.e., Landing-Site Distributions). Finally, at the very bottom of the file, several word-based means are shown for each of the five frequency classes of words. These additional measures are formatted as follows:

Frequency-class means:

Class	FFD	SFD	GD	Pr1	Pr2+	PrS
1	252.43	260.96	297.60	0.697	0.152	0.147
2	243.01	248.36	281.76	0.686	0.128	0.185
3	229.86	229.86	247.98	0.668	0.057	0.274
4	224.11	224.23	228.27	0.479	0.009	0.512
5	217.81	218.22	221.40	0.373	0.004	0.623

The seven columns correspond to the following: (1) Class = frequency class (1 = 1-10; 2 = 11-100; 3 = 101-1,000; 4 = 1,001-10,000; 5 = 10,001+); (2) FFD = first-fixation duration; (3) SFD = single-fixation duration; (4) GD = gaze duration; (5) Pr1 = probability of making one fixation; (6) Pr2+ = probability of making two or more fixations; and (7) PrS = probability of skipping.

2. Setting Up Sentence/Text Files

The sentence/text file should contain four columns of information: (1) each word's frequency of occurrence in printed text; (2) each word's (actual) length in character spaces; (3) each word's mean within-sentence predictability (as tabulated by cloze-task experiments); and (4) a copy of the actual word. The last word of each sentence should also be followed by an ampersand (i.e., @), as indicated in the example below. Without this marker, the model program will assume that all of the words in the file are one big sentence, which may or may not be useful. (For more information about the Schilling et al., 1998 sentence corpus, see Reichle et al., 1998.) Here is an example of how the sentence/text file should be formatted:

1	7	0.00	Nancy's
90	7	0.00	kitchen
9816	3	0.40	was
1	8	0.00	infested
7289	4	1.00	with
6	9	0.00	carpenter
1	4	0.90	ants
28850	3	0.80	and
1	8	0.25	roaches.@

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 file correctly using the *WordIVs* mode before you actually run any real simulations. Also, the sentence/text file should be an ascii file (i.e., a file that only contains alphanumeric characters, and no hidden control characters.)

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

Best regards,
Erik