THE COMPUTATIONAL IMPEMENTATION OF THE LANDSCAPE MODEL

VERSION 1.0

USER'S MANUAL

Paul van den Broek

Yuhtsuen Tzeng

Panayiota Kendeou

CONTACT INFORMATION:

Paul van den Broek
University of Minnesota
211 Burton Hall
178 Pillsbury Drive
Minneapolis, MN 55455
E-mail: pvdbroek@umn.edu

Yuhtsuen Tzeng
National Chung Cheng University
E-mail: ttcytt@ccu.edu.tw

Panayiota Kendeou
University of Minnesota
75 East River Rd
314 Elliott Hall
Minneapolis, MN 55455
E-mail: kend0040@umn.edu

Last Updated: 17 December, 2004

TABLE OF CONTENTS

Page

INTRODUCTION

The complexity of text comprehension demands a computational approach to describe the cognitive processes involved. The computational implementation of the Landscape Model of reading was designed to meet that need. This model captures both on-line comprehension processes during reading and the off-line memory representation after reading is completed, incorporating both memory-based and coherence-based mechanisms of comprehension. Several studies comparing computational and behavioral data indicate that the implemented model is able to account for cycle-by-cycle comprehension processes and memory for a variety of text types and reading situations. For a detailed description of the model's architecture, basic settings, and applications we recommend the paper below. Please, contact us if you need additional information.

van den Broek, P., Young, M., Tzeng, Y., & Linderholm, T. (1999). The landscape model of reading. In H. van Oostendorp & S. R. Goldman (Eds.), *The construction of mental representations during reading* (pp. 71-98). Mahwah, NJ: Erlbaum.

Linderholm, T., Virtue, S., Tzeng, Y., & van den Broek, P. (2004). Flunctuations in the availability of information during reading: Capturing cognitive processes using the Landscape Model. *Discourse Processes, 37*, 165-186.

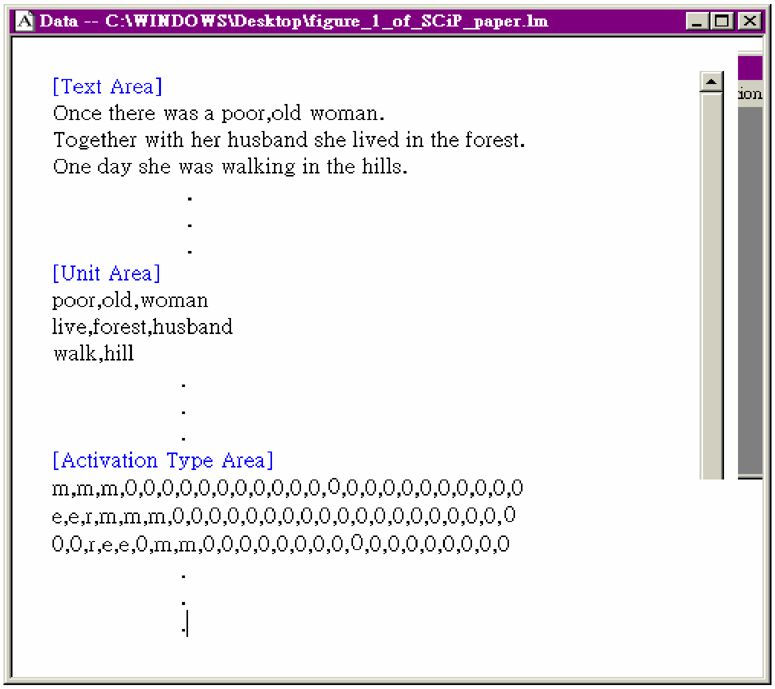
van den Broek, P., Rapp, D. N., & Kendeou, P. (2005). Integrating memory-based and constructionist approaches in accounts of reading comprehension. *Discourse Processes, 39*, 299-316.

SYSTEM REQUIREMENTS AND INSTALLATION

The program designed with C++ programming language could run on Microsoft Windows 95, 98, ME, XP and 2000 with 64MB (minimum) of RAM. Users who have obtained the program can click on the Setup.exe file and the package will install the program automatically and generate the Landscape folder in the Program Files folder with the executive file, Landscape.exe, and several demonstration files – Demo1.lm, and Demo2.lm. For now, the interface of the program is in English but it will take any languages as model inputs as long as those languages are supported by Microsoft Windows.

THE MODEL INPUT

The input to the program consists of a single file that includes three components that should be labeled [Text Area], [Unit Area], and [Activation Type Area]—including the square brackets (see example on page 15, Demo1.lm). These are the three components that the Landscape Model reads and simulates (Figure 1). We describe each component in detail below.



```
[Text Area]
Once there was a poor,old woman.
Together with her husband she lived in the forest.
One day she was walking in the hills.
              .
              .
              .
[Unit Area]
poor,old,woman
live,forest,husband
walk,hill
              .
              .
              .
[Activation Type Area]
m,m,m,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
e,e,r,m,m,m,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,r,e,e,0,m,m,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
              .
              .
              .
```

Figure 1. The Model Input

I. Text Area

In the [Text Area] component, users can type any text without affecting the results of the simulation. Users can type in memos, comments, or (as we recommend) the text they use for the simulation.

II. Unit Area

In the [Unit Area] component, users provide the cue words for the individual parsed text units from the text, with each cue word within a single input cycle separated by a comma. Note that users can decide the unit of analysis in each cycle according to the theory under consideration (e.g., propositions, events, idea units, sentences, and so on). In the example discussed in this manual we use major propositions as text units but users can define text units according to their theoretical purposes (e.g., concepts, main ideas, sentences, and so on). Reading cycles should be separated by hitting the ENTER key. The final cue word in a reading cycle does not need a comma. This format must be followed strictly. Each row of this section is taken as a reading cycle and each cue word within the row is taken as a proposition within a cycle. For a text with *n* cycles, readers should type in *n* rows in this area with each row consisting of its constituent cue words. The Landscape program will use this input to decide the number of reading cycles for the current text and the number of propositions within each cycle. In addition, these cue words will be used as output labels.

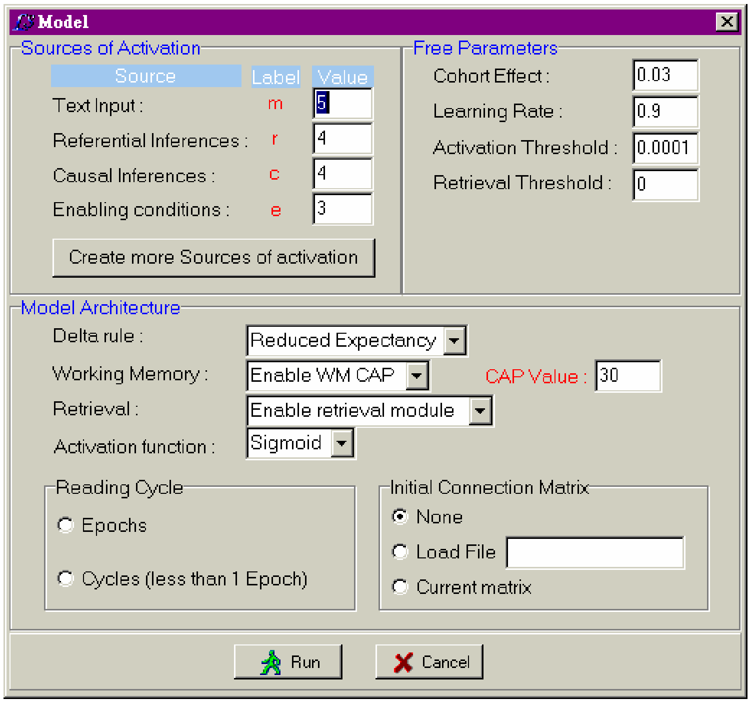
III. Activation Type Area

In the section [Activation Type Area], provide the types of activation each unit receives. These types of activation are labeled using a single alphabetic symbol or a string of alphabetic symbols (e.g., ‘c’ or ‘cau’ for causal, ‘r’ or ‘ref’ for referential, and so on). Units that are not activated in a given cycle are assigned a zero ‘0’. As in the [Unit Area] section, each type of activation unit must be separated by a comma and each reading cycle by the ENTER key. Units at the end of each reading cycle do not need a comma. The number of reading cycles in the [Activation Type Area] and in the [Unit Area] sections should be the same.

RUNNING THE MODEL

Running the LS Program

When users enter all the required information in the three components [Text Area], [Unit Area], and [Activation Type Area] they may proceed to run the program by selecting Model – Parse on the menu bar. If all the required information has been provided, the program will start running. This is indicated by a 'running man' icon that appears on the menu. Users can proceed to set the model's parameter values by clicking on the 'running man' icon. A menu will appear (Figure 2) that includes four important components: Sources of Activation, Free Parameters, Model Architecture, and Run or Cancel. We describe the function of each component below.



Model

Sources of Activation

Source	Label	Value
Text Input :	m	5
Referential Inferences :	r	4
Causal Inferences :	c	4
Enabling conditions :	e	3

Free Parameters

Cohort Effect :	0.03
Learning Rate :	0.9
Activation Threshold :	0.0001
Retrieval Threshold :	0

Model Architecture

Delta rule : Reduced Expectancy

Working Memory : Enable WM CAP   CAP Value : 30

Retrieval : Enable retrieval module

Activation function : Sigmoid

Reading Cycle

○ Epochs

○ Cycles (less than 1 Epoch)

Initial Connection Matrix

● None

○ Load File

○ Current matrix

Figure 2. Running the Model

I. Sources of Activation

The program contains several default types of activation (referential, causal, and enabling connections); the activation values associated with these default sources are based on prior research. However, users can modify the values for their own theoretical purposes, as well as add other types of activation.

If there are any undefined activation variables, and you attempt to run the program you will get an error message *"Some user-defined Activation Values have not been set!"* The program will pick out the undefined activation source variables and ask you to define their activation values one by one. All you need to do is to type in the numerical value that represents the activation value for each variable name (if no values are desired, hit 'OK' and the model will assume zero values). To avoid this error you may hit the “Create more Sources of activation” button and the program will prompt you to define the sources of activation that are undefined.

II. Free Parameters

The program contains several default values for Cohort Effect, Learning Rate, Activation and Retrieval Threshold that are based on prior research. The Cohort Effect parameter determines the cohorts that are formed among text units; larger cohort parameter values produce bigger activation cohort effects. During the process of building cohorts in readers’ working memory, text unit interconnections are built through a delta learning rule. This learning mechanism follows an asymptotic curve as reading elements co-occur more than once in working memory. The amount of learning for each co-occurrence among reading elements is determined by the Learning Rate parameter; larger learning rate parameter values mimic faster learning rate. Activation Threshold is a very low activation value that is taken by the program to decide whether this amount of activation means no activation at all and, therefore, the program will end a loop. It has no particular theoretical interest and is chosen arbitrarily. Retrieval Threshold sets the minimum connection value between text units that can be retrieved during the process of retrieval. It determines the number of text units that will be retrieved and, therefore, mimics readers’ criteria of retrieving.

All free parameter values can be modified according to the user’s theoretical purposes, but we recommend using the default ones unless there is a strong theoretical reason to do otherwise. For example, users may model slow or fast readers by changing the value for the Cohort or Learning Rate parameters accordingly.

III. Model Architecture

The program contains several default values in this section as well. In the box of Delta rule, users have the choice of reduced versus normal expectancy. Expectancy is a property of the delta learning rule; reduced expectancy results in a more stable performance of the model. Users also have choices as to whether to enable working memory capacity constraint and the retrieval component. Once a user enables the working memory module, a choice of capacity value can be made. The Activation function also gives users a choice between a sigmoid or a linear function for the learning curve. As noted above, unless there are strong theoretical reasons to do otherwise we recommend using the default options for each of these parameters.

Another important feature in this section is the Reading Cycles options. By selecting Reading Cycles, the user will be prompted to enter the number of cycles that are read. By changing this value, readers can model reading of a subset of all possible cycles. For example, there are 12 reading cycles in the sample text. If one enters 10 in this menu, the model will read only the first 10 cycles. If users intend to model rereading of the same text for several times, they need to select the Epochs. Users will be prompted to enter the number of epochs. One epoch corresponds to rereading once, two epochs to rereading twice, and so on.

The final feature in this section is the Initial Connection Matrix option. This option allows the simulation of rereading or of reading of multiple texts. Through this option, the simulation will take a connection matrix as an input and build a new memory representation using the pre-existing connection matrix as a starting point. As a result, the program allows users to open an old connection matrix and take it as an input. When users select ‘Current matrix’ they can simulate rereading of the same text. When users select ‘Load File’ they can specify as an input any other matrix. The only limitation here is that the initial matrix must contain all concepts from both matrices, even those set to zero.

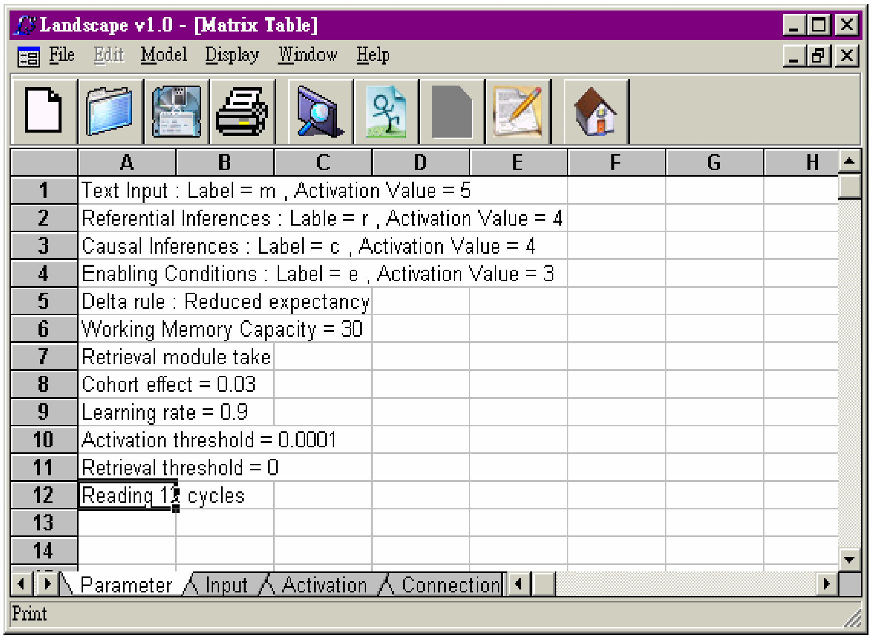
IV. Run or Cancel

After setting the values in the Model menu, the user hits ‘Run’ to run the program. The program will take only a few seconds to display the results. If one wishes to cancel, simply hit ‘Cancel’.

THE MODEL OUTPUT

Looking at the Results

The program gives five output files in an Excel Workbook format. These files are labeled: Parameter, Input, Activation, Connection, and Retrieval (Figure 3).



	A
1	Text Input : Label = m , Activation Value = 5
2	Referential Inferences : Lable = r , Activation Value = 4
3	Causal Inferences : Label = c , Activation Value = 4
4	Enabling Conditions : Label = e , Activation Value = 3
5	Delta rule : Reduced expectancy
6	Working Memory Capacity = 30
7	Retrieval module take
8	Cohort effect = 0.03
9	Learning rate = 0.9
10	Activation threshold = 0.0001
11	Retrieval threshold = 0
12	Reading 12 cycles
13	
14	

Figure 3. The Model Output

I. Parameter

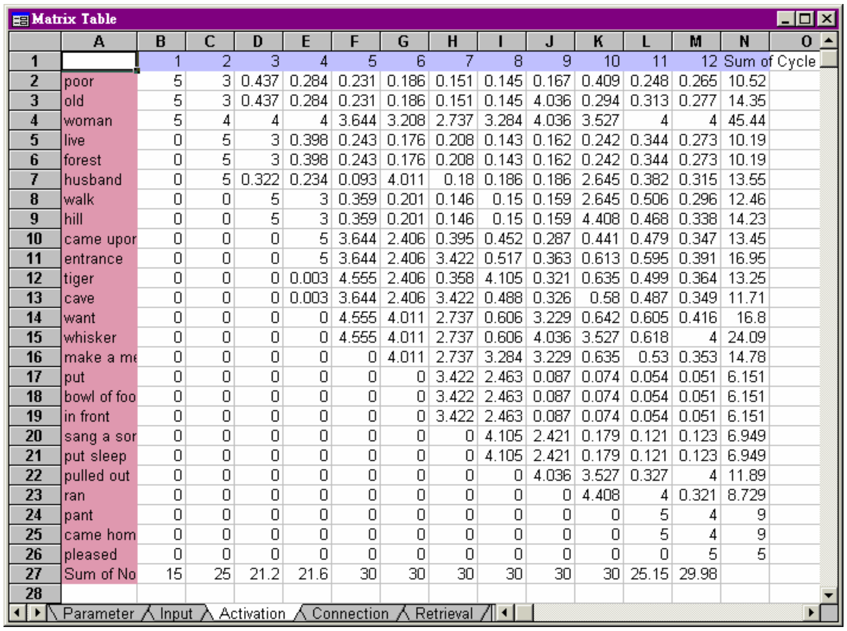
In this file, users will find all the information regarding the settings of the simulation. These include the text input, the types of connections and their values, the delta rule, working memory capacity, whether the retrieval mode was enabled or not, cohort effect, learning rate, and the number of reading cycles. This file is important as a reference for the user.

II. Input

In the input file users will find the input matrix. The input matrix is a matrix that includes all the text units and their values as defined by the user. The first column contains all the cue words in the order they were entered in the [Unit Area]. The other columns correspond to each one of the reading cycles in the order of input. For example, in the first cycle the text units ‘poor’, ‘old’, and ‘woman’ were mentioned. Because ‘m’ mentioning was given the value ‘5’ in the model, these values will appear for each of these three concepts.

III. Activation

In the activation output users will find a matrix table. This table includes all the text units and the activation values they received in each reading cycle. This matrix shows comprehension as it unfolds, online (Figure 4) and differs from the Input matrix that simply depicts types of activation.



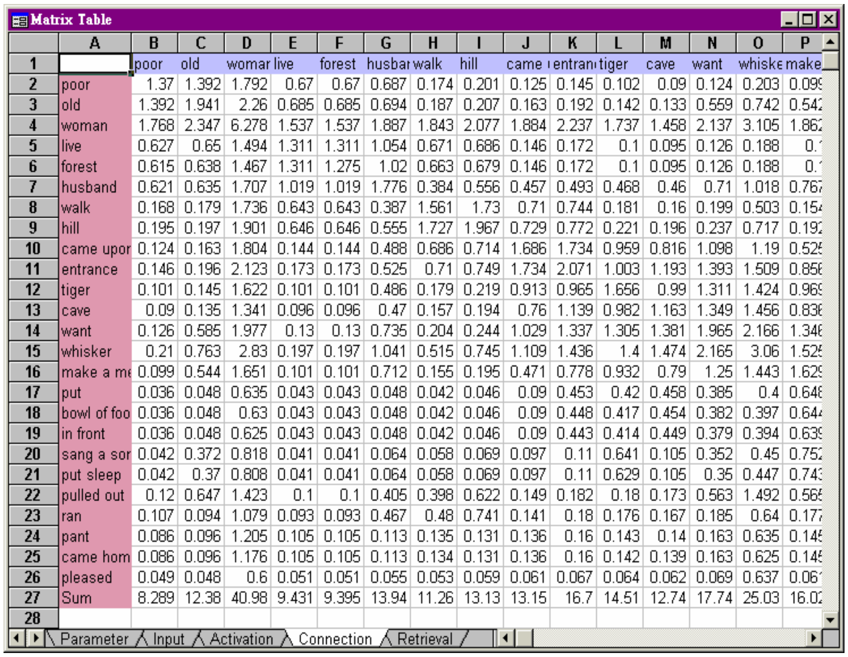
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		1	2	3	4	5	6	7	8	9	10	11	12	Sum of Cycle	
2	poor	5	3	0.437	0.284	0.231	0.186	0.151	0.145	0.167	0.409	0.248	0.265	10.52	
3	old	5	3	0.437	0.284	0.231	0.186	0.151	0.145	4.036	0.294	0.313	0.277	14.35	
4	woman	5	4	4	4	3.644	3.208	2.737	3.284	4.036	3.527	4	4	45.44	
5	live	0	5	3	0.398	0.243	0.176	0.208	0.143	0.162	0.242	0.344	0.273	10.19	
6	forest	0	5	3	0.398	0.243	0.176	0.208	0.143	0.162	0.242	0.344	0.273	10.19	
7	husband	0	5	0.322	0.234	0.093	4.011	0.18	0.186	0.186	2.645	0.382	0.315	13.55	
8	walk	0	0	5	3	0.359	0.201	0.146	0.15	0.159	2.645	0.506	0.296	12.46	
9	hill	0	0	5	3	0.359	0.201	0.146	0.15	0.159	4.408	0.468	0.338	14.23	
10	came upor	0	0	0	5	3.644	2.406	0.395	0.452	0.287	0.441	0.479	0.347	13.45	
11	entrance	0	0	0	5	3.644	2.406	3.422	0.517	0.363	0.613	0.595	0.391	16.95	
12	tiger	0	0	0	0.003	4.555	2.406	0.358	4.105	0.321	0.635	0.499	0.364	13.25	
13	cave	0	0	0	0.003	3.644	2.406	3.422	0.488	0.326	0.58	0.487	0.349	11.71	
14	want	0	0	0	0	4.555	4.011	2.737	0.606	3.229	0.642	0.605	0.416	16.8	
15	whisker	0	0	0	0	4.555	4.011	2.737	0.606	4.036	3.527	0.618	4	24.09	
16	make a me	0	0	0	0	0	4.011	2.737	3.284	3.229	0.635	0.53	0.353	14.78	
17	put	0	0	0	0	0	0	3.422	2.463	0.087	0.074	0.054	0.051	6.151	
18	bowl of foo	0	0	0	0	0	0	3.422	2.463	0.087	0.074	0.054	0.051	6.151	
19	in front	0	0	0	0	0	0	3.422	2.463	0.087	0.074	0.054	0.051	6.151	
20	sang a sor	0	0	0	0	0	0	0	4.105	2.421	0.179	0.121	0.123	6.949	
21	put sleep	0	0	0	0	0	0	0	4.105	2.421	0.179	0.121	0.123	6.949	
22	pulled out	0	0	0	0	0	0	0	0	4.036	3.527	0.327	4	11.89	
23	ran	0	0	0	0	0	0	0	0	0	4.408	4	0.321	8.729	
24	pant	0	0	0	0	0	0	0	0	0	0	5	4	9	
25	came hom	0	0	0	0	0	0	0	0	0	0	5	4	9	
26	pleased	0	0	0	0	0	0	0	0	0	0	0	5	5	
27	Sum of No	15	25	21.2	21.6	30	30	30	30	30	30	25.15	29.98		
28															

Figure 4. The Activation Matrix

Each row shows the history of activation for each text unit across the reading cycles. In this example, the text unit ‘poor’ received ‘5’ on the first cycle, ‘3’ on the second, ‘.437’ on the third, and so on. The last column provides the sum of activation that a unit has received over the course of reading. For example, the unit ‘poor’ received ‘10.52’ total activation over the course of reading of the entire text. Each column represents a reading cycles and shows the activation values each text unit received. In this example, the first reading cycle had a total activation of 15, evenly distributed among three text units, ‘poor’, ‘old’, and ‘woman’.

IV. Connection

In this output users will find a matrix table that includes the strength of the connections between all the text units (Figure 5). These connections create a network between all the text units. Connections between units vary in strength, as indicated by the different connection values in the matrix. In this example, the text unit ‘poor’ is connected with the other text units at various degrees (first row). For instance, ‘poor’ is more strongly connected to ‘old’ (1.392) than to ‘hill’ (.201). The main diagonal is the connection strength value for each concept with itself. For instance, the auto connection strength for ‘old’ is 1.941, for ‘woman’ is 6.278, and so on. These auto connection values are a function of the total activation these units received over the course of reading but also the degree to which they have become connected to other units.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		poor	old	womar	live	forest	husbar	walk	hill	came u	entran	tiger	cave	want	whiske	make
2	poor	1.37	1.392	1.792	0.67	0.67	0.687	0.174	0.201	0.125	0.145	0.102	0.09	0.124	0.203	0.099
3	old	1.392	1.941	2.26	0.685	0.685	0.694	0.187	0.207	0.163	0.192	0.142	0.133	0.559	0.742	0.542
4	woman	1.768	2.347	6.278	1.537	1.537	1.887	1.843	2.077	1.884	2.237	1.737	1.458	2.137	3.105	1.862
5	live	0.627	0.65	1.494	1.311	1.311	1.054	0.671	0.686	0.146	0.172	0.1	0.095	0.126	0.188	0.1
6	forest	0.615	0.638	1.467	1.311	1.275	1.02	0.663	0.679	0.146	0.172	0.1	0.095	0.126	0.188	0.1
7	husband	0.621	0.635	1.707	1.019	1.019	1.776	0.384	0.556	0.457	0.493	0.468	0.46	0.71	1.018	0.767
8	walk	0.168	0.179	1.736	0.643	0.643	0.387	1.561	1.73	0.71	0.744	0.181	0.16	0.199	0.503	0.154
9	hill	0.195	0.197	1.901	0.646	0.646	0.555	1.727	1.967	0.729	0.772	0.221	0.196	0.237	0.717	0.192
10	came upor	0.124	0.163	1.804	0.144	0.144	0.488	0.686	0.714	1.686	1.734	0.959	0.816	1.098	1.19	0.525
11	entrance	0.146	0.196	2.123	0.173	0.173	0.525	0.71	0.749	1.734	2.071	1.003	1.193	1.393	1.509	0.856
12	tiger	0.101	0.145	1.622	0.101	0.101	0.486	0.179	0.219	0.913	0.965	1.656	0.99	1.311	1.424	0.969
13	cave	0.09	0.135	1.341	0.096	0.096	0.47	0.157	0.194	0.76	1.139	0.982	1.163	1.349	1.456	0.836
14	want	0.126	0.585	1.977	0.13	0.13	0.735	0.204	0.244	1.029	1.337	1.305	1.381	1.965	2.166	1.346
15	whisker	0.21	0.763	2.83	0.197	0.197	1.041	0.515	0.745	1.109	1.436	1.4	1.474	2.165	3.06	1.525
16	make a me	0.099	0.544	1.651	0.101	0.101	0.712	0.155	0.195	0.471	0.778	0.932	0.79	1.25	1.443	1.629
17	put	0.036	0.048	0.635	0.043	0.043	0.048	0.042	0.046	0.09	0.453	0.42	0.458	0.385	0.4	0.648
18	bowl of foo	0.036	0.048	0.63	0.043	0.043	0.048	0.042	0.046	0.09	0.448	0.417	0.454	0.382	0.397	0.644
19	in front	0.036	0.048	0.625	0.043	0.043	0.048	0.042	0.046	0.09	0.443	0.414	0.449	0.379	0.394	0.639
20	sang a sor	0.042	0.372	0.818	0.041	0.041	0.064	0.058	0.069	0.097	0.11	0.641	0.105	0.352	0.45	0.752
21	put sleep	0.042	0.37	0.808	0.041	0.041	0.064	0.058	0.069	0.097	0.11	0.629	0.105	0.35	0.447	0.743
22	pulled out	0.12	0.647	1.423	0.1	0.1	0.405	0.398	0.622	0.149	0.182	0.18	0.173	0.563	1.492	0.565
23	ran	0.107	0.094	1.079	0.093	0.093	0.467	0.48	0.741	0.141	0.18	0.176	0.167	0.185	0.64	0.177
24	pant	0.086	0.096	1.205	0.105	0.105	0.113	0.135	0.131	0.136	0.16	0.143	0.14	0.163	0.635	0.145
25	came hom	0.086	0.096	1.176	0.105	0.105	0.113	0.134	0.131	0.136	0.16	0.142	0.139	0.163	0.625	0.145
26	pleased	0.049	0.048	0.6	0.051	0.051	0.055	0.053	0.059	0.061	0.067	0.064	0.062	0.069	0.637	0.061
27	Sum	8.289	12.38	40.98	9.431	9.395	13.94	11.26	13.13	13.15	16.7	14.51	12.74	17.74	25.03	16.02
28																

Figure 5. The Connection Matrix

V. Retrieval

In this file, the program suggests a retrieval order of the text units based on a two-stage retrieval mechanism. The program retrieves the first text unit based on the sum of their total activation values across all reading cycles. Starting from this retrieved text unit, the program searches for and retrieves the text unit with largest connection strength with the first retrieved unit. This searching and retrieving process iterates until it exhausts all the text units or it reaches the retrieval threshold. The output gives two columns. The first column includes the text unit whereas the second column includes their connection value.

SAVING THE RESULTS

We recommend saving the results in the Excel format and opening the files in Excel for further analysis. To save the input file with extension .lm (such as the one in Figure 1), users need to select File at the menu bar and choose Save As to specify a file name. Users can open this file for later editing. To save or edit results of the LS model (such as input matrix, activation matrix, the resulting connection matrix, or the retrieval matrix), users can use the Launch Worksheet Designer under the Display function of the Menu bar. This will work like Excel spreadsheet.

OTHER FEATURES OF VERSION 1.0

This version of LS model is equipped with several additional features (to give suggestions for additional features, please contact the authors).

Reading Delay

The model has ‘delay’ capabilities that allow one to simulate the hypothetical situation that readers wait for a while before continuing reading the next cycle. The program allows users to simulate this by attaching an integer enclosed by a pair of parenthesis at the end of a particular reading cycle (without a comma between the last proposition/event and the parenthesis) under the [Activation Type Area] section. The integer will qualitatively mimic the waiting time interval and this integer will make the program insert a reading cycle with zero activation (see Example 2; file Demo2.lm on page 16). Alternatively, users can simply insert several rows of zero activation values at the proper places in the .lm file and run the program. This might have an impact on the results of the final mental representation of the text. If a delay is inserted between text statements it simulates reflection on the last information in the last input cycle; if is inserted at the end of the text it simulates the gradual decay of the final vector after completing reading of the story.

Rereading

Additionally, this version of the LS program has rereading or multiple text capabilities. This means that the program will take a connection matrix as an input and build a new memory representation on top of the pre-existing connection matrix. As a result, the program allows users to open an old connection matrix and take it as an input and reread. Conversely, the LS program allows users to save a connection matrix easily for later rereading or reading of a new text. The only limitation is that the initial matrix must include all concepts that will appear in the current matrix, even if they have to be set to zero.

EXAMPLE 1 (Demo1.lm)

```
[Text Area]

Once there was a poor, old woman.
Together with her husband she lived in the forest.
One day she was walking in the hills.
She came upon the entrance to a tiger's cave.
She wanted one of the tiger's whiskers.
She wanted to make a medicine of the whisker for her husband.
She put a bowl of food in front of the entrance to the cave.
She softly sang a song which put her the tiger to sleep.
The old Woman quickly pulled out one of the whiskers.
Very quickly, she ran down the hill.
Panting, she came home.
She was very pleased.

[Unit  Area]

poor,old,woman
live,forest,husband
walk,hill
came upon,entrance,tiger,cave
want,whisker
make a medicine
put,bowl of food,in front
sang a song,put sleep
pulled out
ran
pant,came home
pleased

[Activation Type Area]

m,m,m,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
e,e,r,m,m,m,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,r,e,e,0,m,m,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,c,0,0,0,e,e,m,m,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,r,0,0,0,0,0,c,c,m,r,m,m,0,0,0,0,0,0,0,0,0,0,0
0,0,c,0,0,m,0,0,e,e,e,e,m,m,m,0,0,0,0,0,0,0,0,0,0
0,0,r,0,0,0,0,0,0,m,0,m,c,r,c,m,m,m,0,0,0,0,0,0,0
0,0,c,0,0,0,0,0,0,0,m,0,0,0,r,e,e,e,m,m,0,0,0,0,0
0,m,m,0,0,0,0,0,0,0,0,0,c,m,r,0,0,0,e,e,m,0,0,0,0
0,0,c,0,0,e,e,m,0,0,0,0,0,c,0,0,0,0,0,0,c,m,0,0,0
0,0,r,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,r,m,m,0
0,0,c,0,0,0,0,0,0,0,0,0,0,c,0,0,0,0,0,0,c,0,r,r,m
```

EXAMPLE 2 (Demo2.lm)

```
[Text Area]

Once there was a poor, old woman.
Together with her husband she lived in the forest.
One day she was walking in the hills.
She came upon the entrance to a tiger's cave.
She wanted one of the tiger's whiskers.
She wanted to make a medicine of the whisker for her husband.
She put a bowl of food in front of the entrance to the cave.
She softly sang a song which put her the tiger to sleep.
The old Woman quickly pulled out one of the whiskers.
Very quickly, she ran down the hill.
Panting, she came home.
She was very pleased.

[Unit  Area]

poor,old,woman
live,forest,husband
walk,hill
came upon,entrance,tiger,cave
want,whisker
make a medicine
put,bowl of food,in front
sang a song,put sleep
pulled out
ran
pant,came home
pleased

[Activation Type Area]

m,m,m,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0(3)
e,e,r,m,m,m,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,r,e,e,0,m,m,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,c,0,0,0,e,e,m,m,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,r,0,0,0,0,0,c,c,m,r,m,m,0,0,0,0,0,0,0,0,0,0,0
0,0,c,0,0,m,0,0,e,e,e,e,m,m,m,0,0,0,0,0,0,0,0,0,0
0,0,r,0,0,0,0,0,0,m,0,m,c,r,c,m,m,m,0,0,0,0,0,0,0(5)
0,0,c,0,0,0,0,0,0,0,m,0,0,0,r,e,e,e,m,m,0,0,0,0,0
0,m,m,0,0,0,0,0,0,0,0,0,c,m,r,0,0,0,e,e,m,0,0,0,0
0,0,c,0,0,e,e,m,0,0,0,0,0,c,0,0,0,0,0,0,c,m,0,0,0
0,0,r,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,r,m,m,0
0,0,c,0,0,0,0,0,0,0,0,0,0,c,0,0,0,0,0,0,c,0,r,r,m(3)
```

ERROR MESSAGES

Most of the error messages occur in user's attempt to parse the model. This is because of errors in the proposition or activation value areas.

Error Message: **"Some activation value is not correct!"**
This indicates an error in the [Activation Types Area]. It is possible that you have an extra space, or you miss a space.

Error Message: **"There are not enough numbers of activation values in some cycle!"**
This is the most common error message and it could indicate a number of things:

1. A comma was omitted
2. A commas was inserted at the end of a cycle
3. The number of activation values per cycle is not the same
4. The number of cycles in the [Activation Types Area] does not match the number of cycles indicated in the [Unit Area] or vice versa.
5. Inconsistent space between commas and activation values (e.g., [c, 0,m] instead of [c,0,m])

Error Message **"Some user-defined Activation Values have not been set!"**
This error message will occur when you attempt to run the program and there are undefined activation variables. You will be automatically prompt to define those variables.