

The computational implementation of the landscape model: Modeling inferential processes and memory representations of text comprehension

YUHTSUEN TZENG

National Chung Cheng University, Chia-Yi, Taiwan

PAUL VAN DEN BROEK and PANAYIOTA KENDEOU

University of Minnesota, Minneapolis, Minnesota

and

CHENGYUAN LEE

National Kaohsiung Normal University, Kaohsiung, Taiwan

The complexity of text comprehension demands a computational approach to describe the cognitive processes involved. In this article, we present the computational implementation of the landscape model of reading. 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. The overall architecture and specific parameters of the program are described, and a running example is provided. 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.

Reading comprehension is a very complex cognition that involves many separate cognitive mechanisms and results in different levels of representations (Fletcher, 1994; Fletcher & Chrysler, 1990; Kintsch, Welsch, Schmalhofer, & Zimny, 1990; van Dijk & Kintsch, 1983). Invariably, comprehension is assumed to involve interpretation of the information in the text, the use of prior knowledge to do so, and ultimately, the construction of a coherent representation or picture of what the text is about in the reader's mind (e.g., Applebee, 1978; Gernsbacher, 1990; Graesser & Clark, 1985; Kintsch & van Dijk, 1978; Mandler & Johnson, 1977; Rapp, Gerrig, & Prentice, 2001; Stein & Glenn, 1979; Trabasso, Secco, & van den Broek, 1984). This representation is the foun-

ation on which the reader can build for specific reading purposes and types of comprehension. Furthermore, the reader can use this representation to do such things as retelling the text, applying knowledge gathered from the text, identifying the theme, critically appraising the text, and so on. In this fashion, *comprehension* can mean different things to different people, but it always involves a meaningful representation of the textual information in the reader's mind.

Several theoretical models have been developed in an attempt to illustrate the complexity of comprehension processes and to describe different aspects of the reading process (Graesser, Bertus, & Magliano, 1995). Examples of these models include the construction-integration model (Kintsch, 1988, 1998), the structure-building theory (Gernsbacher, 1990), the event-indexing model (Zwaan, Langston, & Graesser, 1995; Zwaan, Magliano, & Graesser, 1995), memory-based text-processing models (McKoon, Gerrig, & Greene, 1996; O'Brien, Rizzella, Albrecht, & Halleran, 1998), causal network theories (Langston & Trabasso, 1998; van den Broek, 1990), and the landscape model (van den Broek, Ridsen, Fletcher, & Thurlow, 1996; van den Broek, Young, Tzeng, & Linderholm, 1999). Each of these models is an attempt to account for text comprehension as a function of the strategic access of information from long-term memory, the automatic activation of information during moment-by-moment reading, or some combination of the two processes.

This project was supported by Grants SC-91-2413-H-194-022 and NSC-92-2413-H-194-026 from the National Science Council of Taiwan to Y.T., by a Golestan fellowship from the Netherlands Institute for Advanced Study in the Humanities and Social Sciences to P.v.d.B., by a Robert and Corrie Beck Fellowship from the College of Education and Human Development at the University of Minnesota to P.K., and by the Center for Cognitive Sciences at the University of Minnesota through Grant HD-07151 from the National Institute of Child Health and Human Development. We thank Eric-Jan Wagenmakers and two anonymous reviewers for their comments on an earlier version of this article. Correspondence concerning this article should be addressed to P. van den Broek, Department of Educational Psychology, University of Minnesota, 178 Pillsbury Drive SE, Minneapolis, MN 55455 (e-mail: pvdbroek@umn.edu).

Some of these theoretical models have been implemented as computational models. The use of computational models in text comprehension research has many advantages. The complexity of on-line processes and off-line representation makes a computational approach a very useful tool for keeping track of all the components within a model and examining their interactions. Goldman, Golden, and van den Broek (in press) have argued that computational models are useful in text comprehension research for at least three reasons. First, computational models stimulate theory development. When implementing a computational model, researchers need to be explicit about assumptions, mechanisms, processes, and so on. This, in turn, refines and improves their theoretical models. Second, computational models assist in the explanation of surprising behavioral data, because they allow researchers to simulate competing hypotheses and compare the computational data with behavioral data. Finally, computational models support communication among researchers within and across fields.

In this article, we present the computational implementation of the landscape model of reading (van den Broek et al., 1996; van den Broek et al., 1999). The aim of the landscape model is to capture both on-line comprehension processes during reading and the off-line memory representation after reading is completed. In this model, reading is conceived of as a cyclical process, in which propositions (or other units of text) fluctuate in their activation from one cycle to the next. In this article, we present a C++ implementation of the landscape model on a Windows-based interface. Our aim is to describe the architecture of the computational model and its basic settings and operations.

COMPUTATIONAL IMPLEMENTATION OF THE LANDSCAPE MODEL

Architecture

The architecture of the landscape model assumes that a reader proceeds through a text in reading cycles, with each cycle corresponding roughly to the reading of a new sentence or proposition (van den Broek, Rapp, & Kendeou, 2005). Across reading cycles, concepts fluctuate in activation as a function of four sources of information; the text input in the current processing cycle, residual information from the preceding cycle, the current episodic text representation, and the reader's prior knowledge. With the reading of each cycle, particular concepts are activated and added as nodes to the episodic memory representation of the text. If a concept is already part of the text representation and is reactivated, its trace is strengthened. In addition, coactivation of concepts leads to the establishment (or strengthening) of connections between those concepts. These cyclical and dynamically fluctuating activations lead to the gradual emergence of an episodic memory representation of the text, in which textual units and inferences are connected via semantic relations (such as causal and referential links). At each

cycle, the network representation constructed during the preceding cycles influences subsequent activation patterns. Thus, the model captures the fluctuations of concepts during reading, as well as the evolving text representation.

Two types of mechanisms guide access to these sources of activation. The first type is *cohort activation*. The architecture of the model assumes that when a concept is activated during reading, all other concepts concurrently activated become associated with it. Thus, each concept connects with other concepts to form a *cohort*. This process follows a delta learning rule and builds associative memory traces (Gluck & Bower, 1988; McClelland & Rumelhart, 1985, 1986). In turn, when one or more of the individual concepts in a cohort become active, the other concepts are also activated. This mechanism is passive and operates under a limited pool of activation. Thus, cohort activation is memory based and similar to the activation mechanism described by the resonance model (Myers & O'Brien, 1998; O'Brien & Myers, 1999; O'Brien et al., 1998; van den Broek et al., 2005).

The second type of mechanism is *coherence-based retrieval*. Unlike memory-based activations that are based on, for example, featural overlap, coherence-based retrieval is a strategic mechanism by which information is retrieved with the aim of meeting a reader's standards or goals (Linderholm, Virtue, Tzeng, & van den Broek, 2004; van den Broek et al., 2005). Such retrieval can be from the episodic text representation constructed so far, from prior knowledge, or from the text itself (e.g., via look-backs in a text). This mechanism operates under a limited pool of activation that can be distributed over concepts and, unlike cohort activation, is strategic. Thus, coherence-based retrieval is similar to *search/effort after meaning* mechanisms described by the constructionist view of reading (Graesser, Singer, & Trabasso, 1994; Singer, Graesser, & Trabasso, 1994).

A central factor in the model that determines which sources of activation are accessed consists of the *standards of coherence* that the reader maintains during reading (van den Broek, Ridsen, & Husebye-Hartmann, 1995). The architecture of the model allows for the adoption of different types of coherence that a reader may establish (including, but not limited to, referential, causal, temporal, and spatial connections). Reader standards can vary as a function of individual differences, text types, reading goals, and so on (Linderholm & van den Broek, 2002; Narvaez, van den Broek, & Ruiz, 1999; van den Broek, Tzeng, Virtue, Linderholm, & Young, 2001), but for narratives and many other types of text, referential and causal standards of coherence are often central. During reading, a reader's standards can, at times, be met entirely by the information currently activated in the model through cohort activation, whereas in other cases, the reader may need to actively search the episodic text representation and/or prior knowledge to maintain these standards through coherence-based retrieval.

Both memory-based and coherence-based comprehension mechanisms closely mimic those proposed in

other psychological theories and are clearly specified in the landscape model. Thus, both bottom-up and top-down mechanisms are integrated in a single theoretical framework. The computational implementation of the model has a great degree of flexibility by allowing users to manipulate different components and adopt different theoretical models. In addition, the model incorporates several essential features of major theories of comprehension, such as the limited capacity of working memory (Just & Carpenter, 1992). As a result, the computational implementation of the landscape model has considerable potential as a research tool in text comprehension research.

Overview of Basic Settings and Operations

The computational implementation of the landscape model allows users to simulate inferential processes and memory representations of comprehension on the basis of a few identified sources of activation. Users need to parse the text and specify the locations and types of inferences and background knowledge that are hypothesized to be activated according to their theoretical stance or research purpose. The program takes a matrix format file as an input. In this input file, each row contains all activated text units in a processing cycle, and each column consists of the history of activation patterns for each particular unit. The program reads the contents of the input file and converts them into a numerical matrix in which activation values are set by the researcher's theory. Each consecutive vector of this input matrix is treated as a reading cycle by the program, and reading elements coactivated in a reading cycle will form a cohort and

build memory strengths among one another. The on-line activation vectors during reading of a text constitute an activation matrix representing the complete on-line activation patterns. The final result produced by the program is the connection matrix, which contains numerical connection strengths of all text units. This matrix represents how tightly the text units connect together and can be used for testing retrieval effects. The data produced by the simulation are in a format compatible with Microsoft Excel, facilitating data extraction and analysis with spreadsheets or statistical applications.

System Requirement and Installation

The program designed with C++ programming language can run on Microsoft Windows 95, 98, ME, XP, and 2000 with 64 MB (minimum) of RAM. This program can be obtained by contacting the authors of this article or via a Web site (<http://education.umn.edu/EdPsych/Projects/LandscapeModel/default.html>). 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 accept any Windows-supported language as input.

HOW TO USE THE LANDSCAPE MODEL

The Model Input

The input to the program consists of a single file that includes three components that should be labeled [Text

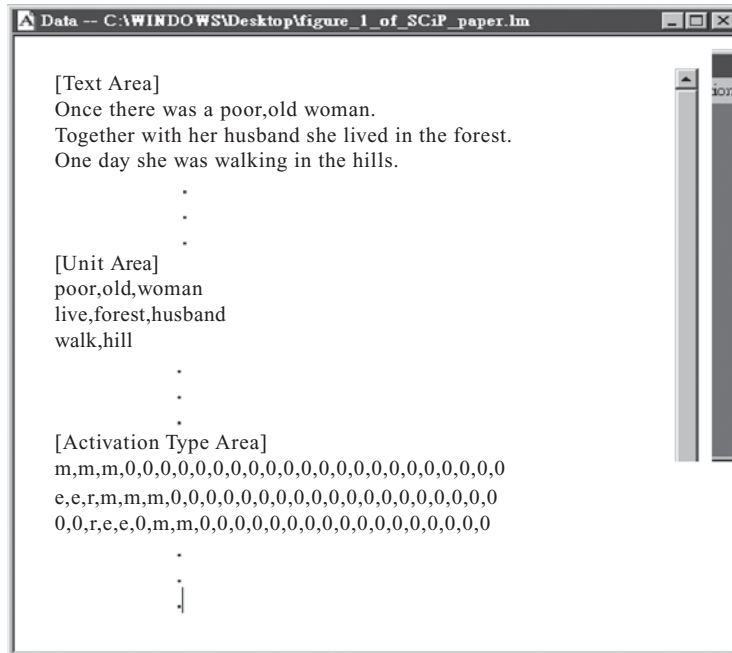


Figure 1. The model input.

Area], [Unit Area], and [Activation Type Area], including the square brackets (see an example in the Appendix). These are the three components that the landscape model reads and simulates (Figure 1). We will describe each component in detail below.

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.

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 article, we use major propositions as the units of analysis. In the program, we denote each major proposition by an individual *cue* word. So, in the first input cycle for the example text (see Figure 1), the words *poor*, *old*, and *woman* represent the propositions poor[old [woman]], old[woman], and [woman], respectively. 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.

Activation type area. In the [Activation Type Area] section, users indicate the type of activation each unit receives. These types of activation are labeled using a single alphabetic symbol or a string of alphabetic symbols. In our example in Figure 1, “m” is used for activation by direct text input (*mention*), “c” for activation by causal connection, “r” for activation by referential connection, and “e” for activation by enabling condition. Units that are not activated in a given cycle are assigned a 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. Also, the total number of units across reading cycles in the [Unit Area] should match the total number of units in each row of the [Activation Type Area]. For instance, there are 2 input items (i.e., *walk*, *hill*) for the third reading cycle in the [Unit Area] but 25 items in the corresponding [Activation Type Area], because the *total* number of units across all the cycles in the entire text, including not-yet-mentioned units, is 25 (in Figure 1, we

show only the 8 units in the first three cycles). As can be seen in Figure 1, in the third reading cycle under the [Activation Type Area], most of the units are not activated, as indicated by zero values. The third unit (*woman*) is activated because it is referentially connected (r), and the fourth and fifth units (*live* and *forest*) are activated because they provide enabling conditions (e). The seventh and eighth units (corresponding to the 2 direct text input items *walk*, *hill*) are activated because they are direct text input items (m). The other units all have zero activation. The complete input file can be found in the Appendix.

Running the Model

After entering the required information in the three components [Text Area], [Unit Area], and [Activation Type Area], users can 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 will describe the function of each component below.

Sources of activation. As can be seen in Figure 2, the program contains several default types of activation (direct text input, through referential inferences, through causal inferences, and by providing enabling conditions). The default activation values associated with these sources are on a relative scale and are based on prior research (van den Broek et al., 1996; van den Broek et al., 1999). However, users can modify these values for their own theoretical purposes and can also add other types of activation by clicking on the “Create more Sources of activation” icon.

Free parameters. The program contains default values (based on prior research) for Cohort Effect, Learning Rate, Activation Threshold, and Retrieval Threshold. 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 rates. Activation Threshold is a value used by the program as a lower threshold of activation, to avoid a computational explosion of minute residual activation values. It has no particular theoretical interest and is chosen at a very low level. Retrieval Threshold sets the minimum connection value between text units that drives the process of retrieval. It determines the number of text units that are retrieved and, therefore, mimics readers’ criteria of retrieving.

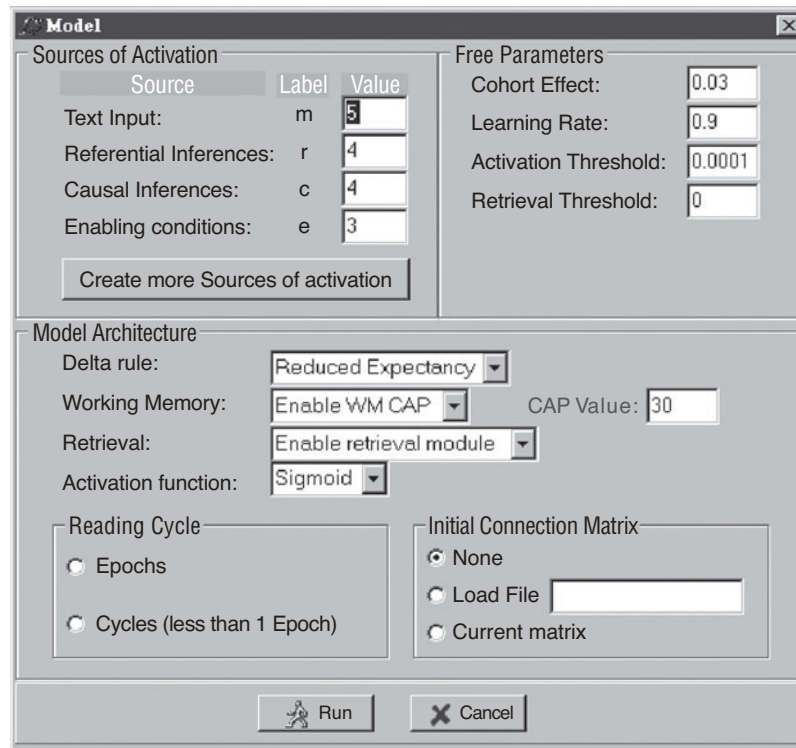


Figure 2. Running the model.

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.

Model architecture. The program contains several default values that determine the model's architecture. (1) The Delta Rule is an implementation of the landscape model's learning mechanism derived from learning theories (Gluck & Bower, 1988; McClelland & Rumelhart, 1985, 1986) and allows users to choose reduced or normal expectancy. Expectancy is a property of the delta learning rule; reduced expectancy results in a more stable performance of the model. (2) Enabling Working Memory capacity allows users to select a limited working memory or attentional capacity value. (3) The Retrieval function allows users to enable a retrieval mechanism that can be used to simulate recall and so on (note that the retrieval mechanism is still under development and, hence, should be considered experimental). (4) The Activation function gives users a choice between a sigmoid and a linear function for the learning curve. This option allows users to compare the relative utilities of these two functions. The sigmoid learning function typically is preferred in computational models because it appears to capture more of the associative learning mechanisms by reaching an asymptote whenever there is repetitive coac-

tivation of text units. As was 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 Cycle options. By selecting 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 (see the Appendix). If the user 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 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 preexisting 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 text units from both matrices, even those set to zero.

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 a user wishes to cancel, he or she needs to hit “Cancel.”

The Model Output

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

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, the cohort effect, the learning rate, and the number of reading cycles. This file is important as a reference for the user.

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 in which 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 directly mentioned (m). Because direct text input is given the value 5 in the default model, these values will appear for each of these three concepts in this cycle.

Activation. In the activation output, users will find a matrix table. This table includes all text units and the activation values they received in each reading cycle. This matrix shows comprehension as it unfolds, on line (Figure 4), and differs from the input matrix, which depicts only types of activation and the associated values.

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, 0.437 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 the entire text. Each column represents a reading cycle 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*.

Connection. In this output, users will find a matrix table that includes the strength of the connections among all text units (Figure 5). These connections create a network between all 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* (0.201). The main diagonal is the connection strength value for each concept with itself. For instance, the autoconnection strength for *old* is 1.941, for *woman* is 6.278, and so on. These autoconnection values are a function of the total activation these units received over the course of reading, as well as of the degree to which they have become connected to other units.

Retrieval. In this output, the program suggests a retrieval order of the text units based on a two-stage retrieval mechanism. The program retrieves the first text unit on the basis of the sum of the total activation values across all reading cycles. Starting from this retrieved text

The screenshot shows a window titled "Landscape v1.0 - [Matrix Table]" with a menu bar (File, Edit, Model, Display, Window, Help) and a toolbar. The main area is a spreadsheet with columns A through H and rows 1 through 14. The data in the spreadsheet is as follows:

	A	B	C	D	E	F	G	H
1	Text Input : Label = m , Activation Value = 5							
2	Referential Inferences : Label = 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								

At the bottom of the window, there are navigation buttons and a status bar with the text "Print".

Figure 3. The model output.

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
2	poor	5	3	0.437	0.284	0.231	0.186	0.151	0.145	0.167	0.409	0.249	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	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 upon	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 m	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 son	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	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.

unit, the program searches for and retrieves the text unit with the 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 its connection value. Alternative forms of retrieval mechanisms are currently being developed.

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 landscape model (such as the input matrix, the activation matrix, the resulting connection matrix, or the retrieval matrix), they can use “Launch Worksheet Designer” under the “Display” function of the Menu bar. This will work like an Excel spreadsheet.

Other Features of the Model

The model has *reading delay* capabilities that allow users to simulate the hypothetical situation in which a reader waits for a while before continuing to read the next cycle—for example, to reflect on the current sentence. The program allows users to simulate this situation by attaching an integer enclosed by a pair of parentheses at the end of a particular reading cycle (without a comma between the last text unit 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 no new input. Higher integers would result in longer waiting times. Alternatively, users can simply insert several rows of zero activation values at the proper places in the .lm file and run the program. Depending on the resulting change in activation and connection patterns, the insertion of delays may 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 it is inserted at the end of the text, it simulates the gradual decay of the final vector after reading of the story is completed.

CONCLUSION

In prior research, the computational implementation of the landscape model has been validated by comparing simulations and human data for the reading of narratives. With regard to on-line measures, the model generates patterns of activation of propositions over the course of reading that can be used to predict the degree of accessibility of those text elements at each point. Indeed, the model’s predicted activation values have been found to be strongly related to the activations reported by human readers ($r = .73, p < .01$; van den Broek et al., 1999). With regard to off-line measures, the model produces memory traces of text propositions that allow prediction of what information is likely to be recalled. In several studies, predicted recall of text propositions has been

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		poor	old	woman	live	forest	husband	walk	hill	came upon	entrance	tiger	cave	want	whisker	make a m
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	0.096
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	1.984	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 upon	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	0.522
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	0.858
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	0.838
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	1.348
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	1.522
16	make a m	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 food	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 song	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	0.568
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	0.148
25	came home	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	0.148
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.

found to be strongly related to actual recall by human readers (r values = .53–.69, all p values < .01; Kendeou & van den Broek, 2004; van den Broek, Kendeou, Sung, & Chen, 2003). In addition to frequency, the model predicted the order of recall by human participants, with the most strongly represented concepts being recalled first and the strengths of semantic relations determining subsequent recall ($R^2 = .49$; van den Broek et al., 1999). Thus, the computational implementation of the landscape model has been found to capture important aspects of the cognitive processes that take place during reading of narratives and of the resulting representation of these texts in memory. The strong correspondence between predicted and observed data indicates that the model captures comprehension processes and memory for texts in a wide range of reading situations and, therefore, has considerable psychological validity.

Although the model initially was implemented to simulate the most commonly investigated phenomena in narrative reading, it has since also been applied to other text genres and reading situations. For instance, the model has been used successfully to simulate expository texts and, within expository texts, refutation texts (van den Broek et al., 2003). The model has also been used to simulate more subtle effects in narrative reading, such as the effects of the emotional value of text statements (Molinari, 2004), of an individual's purpose for reading, of the detection of inconsistencies in a text (Linderholm et al., 2004), and of readers' prior knowledge (Kendeou & van

den Broek, 2004). Furthermore, the model has been applied to contrast different theoretical frameworks, such as memory-based and constructionist views of reading (van den Broek et al., 2005).

The scientific investigation of text comprehension can greatly benefit from endeavors to specify models in precise quantitative terms and develop user-friendly software tools to test and validate these models. In general, computational models stimulate theory development, allow tests of competing hypotheses, and facilitate communication (Goldman et al., in press). In the context of text comprehension research, computational models allow one to track the simultaneous and, possibly, interactive impact of multiple factors that are typically investigated in isolation. For example, although there are numerous investigations of different types of coherence (causal, referential, and others), computational implementations such as that of the landscape model allow one to test hypotheses about the relative and combined contributions of each of these types. As a second example, top-down and bottom-up explanations have been proposed as competing accounts of exactly what determines activation during reading and of how coherence is established during reading; by integrating these two approaches into a single framework, the model opens the door for considering the relative, and possibly collaborative, impacts of these two accounts. As a third example, the computational model facilitates the investigation of individual differences in working memory capacity,

the importance of various types of coherence, learning curves, and so on. As a final example, computational models such as the landscape model allow one to investigate the impact of on-line factors and their interactions on the resulting off-line memory representation.

The computational implementation of the landscape model provides a user-friendly tool for modeling comprehension and testing various hypotheses, because it has many unique features that can shed light on different aspects of comprehension processes, in reading as well as in other communicative settings. In addition, it incorporates important features of several models of comprehension and, therefore, has the potential for implementing and contrasting different theoretical frameworks.

REFERENCES

- APPLEBEE, A. N. (1978). *The child's concept of a story: Ages two to seventeen*. Chicago: University of Chicago Press.
- FLETCHER, C. R. (1994). Levels of representation in memory for discourse. In M. A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 589-607). San Diego: Academic Press.
- FLETCHER, C. R., & CHRYSLER, S. T. (1990). Surface forms, textbases and situation models: Recognition memory for three types of textual information. *Discourse Processes*, **13**, 175-190.
- GERNSBACHER, M. A. (1990). *Language comprehension as structure building*. Hillsdale, NJ: Erlbaum.
- GLUCK, M. A., & BOWER, G. H. (1988). From conditioning to category learning: An adaptive network model. *Journal of Experimental Psychology: General*, **117**, 227-247.
- GOLDMAN, S. R., GOLDEN, R., & VAN DEN BROEK, P. (in press). Why are computational models of text comprehension useful? In F. Schmalhofer & C. Perfetti (Eds.), *Higher level language processes in the brain*. Mahwah, NJ: Erlbaum.
- GRAESSER, A. [C.], BERTUS, E. L., & MAGLIANO, J. P. (1995). Inference generation during the comprehension of narrative text. In E. P. Lorch & E. J. O'Brien (Eds.), *Sources of coherence in reading* (pp. 295-320). Hillsdale, NJ: Erlbaum.
- GRAESSER, A. C., & CLARK, L. F. (1985). *The structures and procedures of implicit knowledge*. Norwood, NJ: Ablex.
- GRAESSER, A. C., SINGER, M., & TRABASSO, T. (1994). Constructing inferences during narrative text comprehension. *Psychological Review*, **101**, 371-395.
- JUST, A. M., & CARPENTER, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, **99**, 122-149.
- KENDEOU, P., & VAN DEN BROEK, P. (2004, November). *Modeling comprehension of non-narrative genres: Expository and refutation texts*. Paper presented at the 34th Annual Meeting of the Society for Computers in Psychology, Minneapolis.
- KINTSCH, W. (1988). The role of knowledge in discourse comprehension: Construction-integration model. *Psychological Review*, **95**, 163-182.
- KINTSCH, W. (1998). *Comprehension: A paradigm for cognition*. New York: Cambridge University Press.
- KINTSCH, W., & VAN DIJK, T. A. (1978). Towards a model of text comprehension and production. *Psychological Review*, **85**, 363-394.
- KINTSCH, W., WELSCH, D. M., SCHMALHOFER, F., & ZIMNY, S. (1990). Sentence memory: Theoretical analysis. *Journal of Memory & Language*, **29**, 133-159.
- LANGSTON, M. C., & TRABASSO, T. (1998). Modeling causal integration and availability of information during comprehension of narrative texts. In H. van Oostendorp & S. Goldman (Eds.), *The construction of mental representations during reading* (pp. 29-69). Mahwah, NJ: Erlbaum.
- LINDERHOLM, T., & VAN DEN BROEK, P. (2002). The effects of reading purpose and working memory capacity on the processing of expository text. *Journal of Educational Psychology*, **94**, 778-784.
- LINDERHOLM, T., VIRTUE, S., TZENG, Y., & VAN DEN BROEK, P. W. (2004). Fluctuations in the availability of information during reading: Capturing cognitive processes using the landscape model. *Discourse Processes*, **37**, 165-186.
- MANDLER, J. M., & JOHNSON, N. S. (1977). Remembrance of things parsed: Story structure and recall. *Cognitive Psychology*, **9**, 111-151.
- MCCLELLAND, J., & RUMELHART, D. (1985). Distributed memory and the representation of general and specific information. *Journal of Experimental Psychology: General*, **114**, 159-188.
- MCCLELLAND, J., & RUMELHART, D. (1986). A distributed model of human learning and memory. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 2: Psychological and biological models* (pp. 170-215). Cambridge, MA: MIT Press.
- MCKOON, G., GERRIG, R. J., & GREENE, S. B. (1996). Pronoun resolution without pronouns: Some consequences of memory-based text processing. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **22**, 919-932.
- MOLINARI, C. (2004). *Inference of fictional characters' emotions: Implementation in the landscape computational model of reading comprehension*. Unpublished doctoral dissertation, University of Buenos Aires, Argentina.
- MYERS, J. L., & O'BRIEN, E. J. (1998). Accessing the discourse representation during reading. *Discourse Processes*, **26**, 131-157.
- NARVAEZ, D., VAN DEN BROEK, P., & RUIZ, A. B. (1999). The influence of reading purpose on inference generation and comprehension in reading. *Journal of Educational Psychology*, **91**, 488-496.
- O'BRIEN, E. J., & MYERS, J. L. (1999). Text comprehension: A view from the bottom up. In S. R. Goldman, A. C. Graesser & P. van den Broek (Eds.), *Narrative comprehension, causality, and coherence: Essays in honor of Tom Trabasso* (pp. 35-53). Mahwah, NJ: Erlbaum.
- O'BRIEN, E. J., RIZZELLA, M. L., ALBRECHT, J. E., & HALLERAN, J. G. (1998). Updating a situation model: A memory-based text processing view. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **24**, 1200-1210.
- RAPP, D. N., GERRIG, R. J., & PRENTICE, D. A. (2001). Readers' trait-based models of characters in narrative comprehension. *Journal of Memory & Language*, **45**, 737-750.
- SINGER, M., GRAESSER, A. C., & TRABASSO, T. (1994). Minimal or global inference during reading. *Journal of Memory & Language*, **33**, 421-441.
- STEIN, N. L., & GLENN, C. G. (1979). An analysis of story comprehension in elementary school children. In R. O. Freedle (Ed.), *New directions in discourse processing* (Vol. 2, pp. 53-120). Hillsdale, NJ: Erlbaum.
- TRABASSO, T., SECCO, T., & VAN DEN BROEK, P. W. (1984). Causal cohesion and story coherence. In H. Mandle, N. L. Stein, & T. Trabasso (Eds.), *Learning and comprehension of text* (pp. 83-111). Hillsdale, NJ: Erlbaum.
- VAN DEN BROEK, P. (1990). The causal inference maker: Towards a process model of inference generation in text comprehension. In D. A. Balota, G. B. Flores d'Arcais, & K. Rayner (Eds.), *Comprehension processes in reading* (pp. 423-445). New York: Academic Press.
- VAN DEN BROEK, P., KENDEOU, P., SUNG, Y. C., & CHEN, M. (2003, June). *Comprehension and memory of science texts: A simulation using the landscape model*. Poster presented at the Annual Meeting of the Society for Text and Discourse, Madrid.
- 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.
- VAN DEN BROEK, P., RISDEN, K., FLETCHER, C. R., & THURLOW, R. (1996). A "landscape" view of reading: Fluctuating patterns of activation and the construction of a stable memory representation. In B. K. Britton & A. C. Graesser (Eds.), *Models of understanding text* (pp. 165-187). Hillsdale, NJ: Erlbaum.
- VAN DEN BROEK, P., RISDEN, K., & HUSEBYE-HARTMANN, E. (1995). The role of readers' standards of coherence in the generation of inferences during reading. In E. P. Lorch & E. J. O'Brien (Eds.), *Sources of coherence in reading* (pp. 353-373). Hillsdale, NJ: Erlbaum.
- VAN DEN BROEK, P., TZENG, Y., VIRTUE, S., LINDERHOLM, T., &

