The Predication Semantics Model: The Role of Predicate Class in Text Comprehension and Recall

by
Althea A. Turner*
Paul B. **Andreassen****
Bruce K. **Britton*****
Deborah **McCutchen******

Working Paper No. 78

August 1992

* NYNEX Science & Technology, Inc.
** The Jerome Levy Economics Institute
*** Department of Psychology, University of Georgia
**** Department of Educational Psychology, University of Washington

Abstract

Previous models of text comprehension generally do not maintain a coherent set of propositions in working memory. Readers produce coherent summaries of texts, however, even when they lack extensive background knowledge. We present a computational model of text comprehension and recall that makes extensive use of predicate class information, which explains how readers extract the gist of a text. Predicate class determines the sequence in which propositions are reduced, and the rules that are used for gist extraction. A computer simulation of the model significantly predicted subjects' immediate recall and maintained coherence in working memory on most processing cycles.

The primary assertion of this paper is that predicate class contains semantic information that readers use to make generally accurate predictions about the relative importance of a given proposition, and then which arguments to hold, pass to another proposition, or eliminate as they construct a summary representation of the text. We will show how different classes of predicates do more than define the relationship among each proposition's arguments. A causal predicate, for example, does not just assert that one of a proposition's arguments is a cause and that the other is an effect. Such a predicate also suggests the relative importance of the proposition, the relative importance of its two arguments, and the ways in which the proposition may be incorporated with or by other propositions in the possess of summarizing and encoding the text_ The semantic information found in the predicates of a text is an important part of what allows people to grasp the gist of a text even when they have only minimal background knowledge (e.g., Kieras, 1985; Perrig & Kintsch, 1985; Schmalhofer, 1982).

Determining general rules for making use of the semantic information in predicate class is complicated by the fact that predicate semantics is part of a larger context. First, the semantic information contained in predicate class is more likely to lead to coherent summarization of a text if the propositions read into memory on the most recent input cycle form a coherent whole before the summary-producing reduction process is begun. The input process needs to be controlled so that it stops at an appropriate place for the summary producing reduction process to begin. Second, the process of reducing the text to gist needs to avoid creating propositions with undefined arguments. The end product of one cycle of reduction must form a coherent base for the next cycle to occur. Third, predicate class carries only part of the semantic information available to readers to use in the reduction process (van Dijk & Kintsch, 1983; Kintsch & van Dijk, 1978). Domain knowledge also may be required. For example, given the sentence, "John gave Bob a book in the library at noon," a domain independent model based on predicate semantics might generally consider the location and time arguments optional. However, doing so in a courtroom drama or detective story might drop the critical elements of an alibi. Our goal has been to define a set of predicate class based rules that operate across the most general set of conditions.

In this paper, we present and test a model of text comprehension, the predication semantics model, that extends previous case grammar approaches (Filmore, 1968; Gruber, 1965; Jackendoff, 1972). The model incorporates explicit procedural rules that make use of the semantic information carried by predicates. Like previous models, the one presented here works in a series of input and reduction cycles. During the input phase, propositions are brought into a working memory controlled by a dynamic set of stopping rules. These rules automatically balance limits on the number of propositions held in working memory (Kintsch & Keenan, 1973; Miller, 1956), with clause boundary markers (Fodor, Bever & Garrett, 1974) and with internal completeness (de Beaugrande, 1980; Kintsch, 1988). During the reduction phase, the procedural rules extract the gist from the propositions currently in the model's working memory. Information about the propositions in the working memory is transferred to a long-term memory store throughout the reduction phase.

The primary innovations of the predication semantics model come during the reduction phase of the cycle. The model uses a semantic system defined across sets of predicate classes (Turner, 1987; Turner & Greene, 1978) along with the input recency and connectedness of the predicates to set the order in which the propositions are reduced. The rules associated with a given proposition are then used to reduce that proposition to gist, that is, to reduce the amount of information in working memory while preserving as much of the meaning as possible. These rules play the major role in maintaining coherence in working memory.

In the next section, we describe the input and reduction phases in detail. We then test the model on four texts by comparing its predicted long term memory strength values with subjects' recall data. A computer simulation of the model is used to derive predictions of the model's behavior. There are three primary facets of this test. First, we test the model by examining how changing the memory parameters affects the amount of free recall variance accounted for. The model predicts a substantial portion of the variance of subjects' recall, and is robust across a wide range of parameter values. Second, we test the model by looking to see how well it fits the various subcategories of predicates. Most of the residual variation occurs as a result of nonsystematic errors. Finally, we examine the contents of working memory and check whether the model has maintained a coherent set of propositions at the end of each reduction cycle. The model maintains coherence on more than 70 percent of the cycles.

The central purpose here is to explore the extent to which predicate categories play a role in reading comprehension and recall. As such, in formulating the computer simulation used to test the predication-semantics model, we have not included more than a minimum of real world knowledge. Similarly, we have excluded most morpheme level semantic features, all nonexplicit causal networks, any macrotextual valuative time-series structure and other factors that undoubtedly play a role in reading. Moreover, although any case grammar model requires complexity, we have tried to be parsimonious. The simulation involves only five input stopping criteria, one parameter to calculate propositional strength, five categories to sequence reduction, four basic reduction actions, and it uses similar reduction rules across categories of predicate classes. Parsimony has led to oversimplification. Our rule for the reduction of negation predicate propositions, for example, is simply to cut the entire section of the semantic network referenced by the proposition and not referenced elsewhere. Similarly, the calculation of memory strength of a proposition is based solely on the contents of working memory, and does not draw on related information in long term memory. The success of the model stands as evidence of our central hypothesis: that predicate class plays a central role in guiding comprehension and recall of text.

The Predication Semantics Model

Input Phase

The model uses several criteria to locate the end of a connected chunk of information. When these criteria are satisfied, the model stops the current input phase. The model was designed to stop input at the end of a network of connected propositions, which is the point at which propositions in working memory are generally most readily reduced to a coherent gist.

Five criteria are used for ending the input phase. First, an absolute limit is set on the number of propositions that can be read into working memory. Second, the model checks for end of sentence markers. Input generally stops when a period, exclamation point, or question mark is encountered in the text Exceptions are abbreviation marks and one-word sentences. When an end of sentence marker is encountered, the remaining criteria for stopping input are relaxed. Third, end of clause markers such as those marked by commas, parentheses, colons and semi-colons are taken into account. Readers frequently pause at clause boundaries in long sentences. The model avoids stopping during lists of simple arguments. Fourth, the model checks the arguments of the propositions to see if they form connected chains. The ease of extracting a coherent gist of the information in working memory is likely to increase if the information is interconnected before the extraction process begins (Haviland **&** Clark, 1974). As each proposition is read into working memory, its arguments are checked for overlap with those propositions already in working memory. The criteria for stopping input are relaxed whenever all propositions in working memory form connected chains. Finally, the individual propositions are checked for internal completeness. The input model relaxes the stopping criterion when all of the new propositions are internally complete.

The computer simulation of the model works simultaneously from the original text and from a propositional analysis of the text encoded using Turner's (1987) propositional analysis. During the input phase, the simulation reads the text word by word, and consults the proposition list to determine whether the given word was used in a proposition. Every predicate and **all** simple arguments (those not themselves propositions) are anchored during coding to the appropriate word in the text. When the most recently input word is **used in** a proposition, then that part of the proposition is activated and that information becomes available. When a proposition is activated for the first time, then the appropriate position in the proposition is **filled** and blank slots are opened as needed for a missing predicate or for missing arguments. If a proposition is read that uses as an argument a proposition that is no longer in working memory, that proposition can be recalled from long-term memory.

Working memory also holds information about recency and connectedness of the propositions. At the end of the input phase, the propositions in working memory are passed forward to the reduction phase.

Reduction Phase

The reduction phase of the model selectively constructs the gist propositions in working memory. After reduction is complete, these gist propositions are passed to the next cycle. The reduction phase decreases the number of propositions in memory while **maintaining** coherence among them. In constructing the gist propositions, coherence needs to be balanced with **recency to allow the** contents at the end of reduction to serve as a suitable base for the propositions brought in on the upcoming input cycles.

There are three major steps in the reduction phase: ranking, rule application, and reassessment. **Predicate**-semantics dominate the first two. During the ranking procedure, the propositions are categorized in terms of their general contribution to gist. During the rule application procedure, the predicate class of the selected proposition determines the reduction rule that applies. The reduction rules use four basic actions contingent upon the proposition's semantic-predicate class, its context, and the nature of its arguments. If the proposition initially selected as the first candidate cannot be reduced, then the rule application procedure passes to the second candidate, and so on. The reassessment step examines the success of the rules as to whether enough reduction has occurred to free the working memory sufficiently for the next input phase, and whether the contents of the working memory are reasonably well connected Otherwise, the remaining propositions are ranked and the steps repeated.

Ranking

The ranking procedure orders propositions in working memory into five categories. These are propositions with minimal information predicates, old and unconnected propositions, description predicate propositions, coordination predicate propositions, and unclassified predicate propositions.

The first category reduced contains propositions that commonly carry information that contributes minimally to gist. These include propositions that are exact duplicates of other propositions, and those with REFERENCE, HEDGE and INTENSIFY predicates. (Defined predicate-semantic classes are shown in capitals.) REFERENCE predicate propositions provide the referents of pronouns and alternative nominal phrases. They were reduced before propositions that were exact duplicates of those already in working memory, because otherwise the duplication may not become apparent. Resolving the references before the remaining propositions simplifies the reduction process. Predicates from the first category are illustrated in Table 1. Both HEDGE and INTENSIFY proposition classes take two arguments. The word or phrase that provides the hedge, and the word or phrase that provides intensification can generally be deleted without affecting gist. Although in some cases these propositions may seem potentially central to the gist of the contents of working memory (e.g., "The rope almost reached, but then he vanished"), their importance in such cases usually stems from being embedded in a proposition with a coordination predicate. The rules for reducing coordination predicate propositions generally preserve the gist of such HEDGE and INTENSIFY propositions after the latter are reduced.

Insert Table 1 about here

The second reduction category contains propositions that formerly appeared in higher ranked categories, but which have acquired low strength values as a consequence of becoming old and unconnected. The purpose of this category was to keep the contents of working memory relevant Old propositions are those that have endured at least one previous cycle of reduction. New propositions are those read into working memory on the most recent input cycle, and a new argument is one used in a new proposition. A proposition has one connection (a) for every new argument it shares with another proposition, (b) for each instance it embeds a new proposition as an argument, and (c) for each case in which it is itself embedded as an argument. Note that embedding an old proposition as an argument does not count as a connection. This was done so that reduction would be directed toward the most central node of a network of older propositions. Given the evidence concerning given and new information in discourse, the distinction between old and new was made to keep the propositions in working memory at the end of a reduction phase reasonably current with those being brought in on upcoming processing cycles. Connections were presumed to determine the strength of a proposition by binding the proposition more tightly within the activated associative net (Anderson, 1983). (The manner in which strength is computed is discussed in the next section.)

The next two categories hold the remaining predicates of the classification system. The third reduction category contains propositions with description predicates. In these predicate classes, one argument commonly provides information about another, such as what group a person comes from, his or her personality, how much of something was involved, and its composition. Description predicates also mark the beginning of an action, how it was done, and the like. These predicate classes are illustrated in Table 2. The fourth reduction category contains propositions with coordination predicates, which are illustrated in Table 3. Coordination propositions are those predicate classes that generally bind other propositions together. Such propositions include those that identify a speaker and receiver with what was said, identify where an event occurred, and compare one event with another along a particular dimension. Description predicate propositions were set to be reduced before those with coordination predicates because the latter, by binding propositions together, tend to provide the greatest structural coherence to propositions in working memory.

Insert Tables 2 and 3 about here

The fifth and last reduction category contains propositions that are generally protected from reduction. To fall into this category, a proposition must have a predicate not defined in the classification system. This includes about one-third of the propositions in text. These predicates do not have specific reduction rules to transform, abstract, or simplify them. The reason these propositions are protected from reduction is because they tend to include verbs that are unique to a given type of text, such as verbs of movement and change, and such propositions are often central to the gist of the text. Propositions with unclassified predicates are deleted from working memory only when all the propositions embedded in them have been reduced to simple arguments, and when they have no connections to active propositions.

When more than one proposition falls in a given reduction category, the ranking procedure orders them by strength. When strength values are tied, the older proposition is scheduled to be reduced first.

Strength. All three steps of the reduction phase use the strength values of the propositions. During the ranking procedure, these values help order propositions. During the application of the reduction rules, there are rules that use strength to determine whether to fire, or which of a proposition's arguments to reduce. During reassessment, average strength is used in the process that determines whether or not to end reduction.

The strength for a given proposition is determined by three factors: its strength at the beginning of the reduction cycle; its current number of connections; and the model's only parameter, a The strength of a new proposition with no connections equals α. The decay of strength from the end of one cycle to the beginning of the next equals l-a. Finally, each connection increases the strength of the proposition by a constant proportion, a, of the possible strength remaining.' Strength values are recalculated before each reduction step because a proposition's suength can change within a reduction phase by the gain or loss of a connection. This is the only way strength can change within a reduction phase.

The a parameter determines the general proportion of old propositions versus new in working memory. With high values of a (i.e., a ≥ .5), strength depends primarily on recency, and on having connections in the current cycle. Propositions with more current connections are stronger than those with fewer, and with an equal number of connections, new propositions are stronger than old. With low values of a (i.e., a c .2), strength is more a function of connections on any cycle, previous or current A greater sum across the current and previous cycle dominates a lesser sum.

The point at which a proposition drops from the third, fourth, or fifth reduction categories to the second was set to a. As such, low values of a maintain the distinction among description, coordination, and unclassified predicate propositions longer. One consequence is that, with low values of a, propositions with unclassified predicates (i.e. most verbs) are likely to stay in short term memory for more input-reduction cycles.

Rule Application

The reduction procedure uses predicate class to determine which reduction rule to use on a given proposition. Each reduction rule defines a set of actions to attempt, and in some cases, what conditions to set before these actions can take place. The rules vary in complexity. Some rules attempt a sequence of possible steps, returning control to the reassessment procedure if a step occurs, otherwise passing to the next possible step. Other rules call for only one fixed action. All rules face some limitations. For example, no reduction rule can remove a proposition from working memory if it is embedded in another proposition; this helps maintain a connected set of propositions.

Basic actions. There are four basic actions in reduction. First, CORE replaces all references to a proposition with one of its arguments, and then deletes the original proposition from working memory. For example, reducing the proposition expressing the part/whole relationship in “he hit the car’s bumper” leads to “he hit the car.” Of the four basic actions, CORE occurs in more rules than any other; it is used in 25 of the 30 predicate class reduction rules. Rules use CORE when one argument is usually the part of a proposition that makes the central contribution to gist. As can be seen in Table 1, CORE is the only action used for propositions with immediate resolution predicates. CORE BEST is a related rule which uses a strength criterion to select which argument to CORE, as shown for some of the predicates in Tables 2 and 3. Second, REDUCE initiates the reduction of a proposition embedded as an argument_ This action deflects the immediate reduction of the selected proposition to one it uses as an argument. The predicate class of the embedded proposition determines how that proposition is reduced, and REDUCE takes no effect when an embedded argument is not a proposition. As shown in Tables 2 and 3, REDUCE is primarily used in identification and location propositions to remove the context around the less important argument before CORE is applied to the more important one. Third, TRANSFORM changes predicate class to make specific quantitative information more general. Fourth, PRUNE successively deletes propositions nested within a given propositional argument, stopping only if it encounters a proposition that is referenced by another proposition. The original argument itself is then deleted, and if no arguments remain, so is the entire proposition. This rule is employed only when the third rule has made an argument irrelevant or when removing the original proposition might otherwise cause an unintended reversal in the meaning of the gist of the propositions in working memory if the offending propositions were to remain. For example, removing a negation predicate proposition requires that the chain of arguments nested within it also be deleted. Finally, there is an initial condition, a nonaction, used with logical relation propositions. This condition, PROTECT, prevents the first three actions from occurring when a proposition's strength value exceeds l-a. Low values of a effectively disable PROTECT.

Rule construction concerns. Several guidelines were used in constructing the reduction rules in the model simulation. The rules were kept as simple as possible, to accord with people's finite cognitive resources. They were designed for the general case. Reduction rules based on predicate class cannot be error free. Only some of the real world knowledge that readers might use in reduction is contained in predicate class. Reduction was also seen as part of a larger constructive process. The rules were designed to provide not only a coherent set of propositions at the end of a given input cycle, but also to serve as a basis for interpreting propositions input on future cycles. Rather than standing in isolation, the rules rely on the surrounding ranking and reassessment procedures. The reduction rules appear in short hand form in the right columns of Tables 1 to 3.

The influence of these guidelines can clearly be seen by the use of CORE even when more complex conditional action sequences might keep gist coherent under special circumstances. This is not as naive as it might first seem. For example, selecting the whole as the core argument of PART propositions rather than the part argument is commonly functional (e.g., "car" rather than "fender"), but it is generally infelicitous when the text centers on discriminating between parts of a single entity (e.g., "he hit the car's fender and its right-hand door, but missed its back panel"). Some protection against such an untoward move comes from the rule itself. If different parts are being discussed, they often take modifiers to distinguish them (e.g., "the car's front bumper, not its back bumper"), which means that they will appear in a PART proposition embedded as propositions themselves, rather than as simple arguments. The rule delays taking the whole as the core so long as the part is nested as a proposition that can itself be reduced. A successful **REDUCE** action fulfills the one action step per rule quota. Only if no reduction can occur does the rule "otherwise," to use the parlance adopted in Tables 1 to 3, **CORE.** Additional protection is provided by the other reduction steps. When the text centers on discriminating between parts of one entity, the whole is a shared argument. This provides a high initial strength to the various part propositions (e.g., by the repeated sharing of "car"), and the ranking step protects them from reduction. When their strength values fall, they do so in concert The consequence is that they are likely all to be dropped on a single reduction phase, deleting the entire idea. This avoids the infelicity. The overall consideration in the rules is to have them lead toward coherent gist.

Full description and explanation of the reduction rules appears in Appendix A. Briefly, most reduction rules rely on the **CORE** action. With identification predicate class propositions, unessential information is first cleared away. Quantification predicate class propositions and contrasts are reduced by simplifying the arguments, making the information more general. Locative and temporal predicate class propositions are constructed to bind events, times and dates together in memory as well as possible. Logical relations are initially protected, before the central argument is identified.

Reassessment

The third step of a reduction sequence determines whether reduction has been sufficient to end the reduction phase. If insufficient reduction has occurred, the model returns to the ranking step. There are several criteria for ending a reduction phase. Fist, the number of propositions in working memory must fall below a given level. Just as there is a maximum number of propositions that can be brought into working memory during the input cycle, there is a maximum number of propositions that the reduction phase may hold over into the next processing cycle. If this level has been reached, then the program assesses whether the remaining propositions are sufficiently connected. To keep the level of interconnectedness reasonably high, with at least an average of one active connection, this value was set to the strength value halfway between that of a new proposition with one active connection and that of such a proposition on the next cycle if it again has one active connection. This criterion generally serves to produce a coherent network of interconnected propositions. Fiily, there is a lower bound such the reduction rules may not eliminate all of the propositions from working memory, reducing the gist to nothing. As the number of propositions in working memory falls, less interconnectedness may be needed to maintain the existing network (e.g., Norman & Bobrow, 1975).

Long term memory

Long term memory traces are strictly the product of working memory processes. Two basic notions were used to determine how much strength a proposition gains in long term memory each time a later version of it is processed in working memory. First, given limited active cognitive processing resources, the effect of any given transfer to long term memory declines as the number of propositions in working memory is increased (Baddeley & Hitch, 1974). The long term strength value equals a proposition's working memory strength divided by the number of propositions in working memory. The second notion was that memory improves with additional processing. Each transfer to long term memory for a proposition, in either original or reduced form, increases the proposition's long term strength. The model allows propositions to be transferred to long term memory at two points -- at the end of each input phase, and each time a proposition is altered within a cycle. When an altered version is transferred, its strength is added to the original proposition. Propositions accumulate no further strength after elimination from working memory.

The input phase of the model was used to derive two additional memory measures. Previous research has shown that readers remember more information from early in the passage, where the theme is introduced, and from early in the paragraph, whether topic sentences usually occur (Mandler, 1978; Mandler & Johnson, 1977; van Dijk & Kintsch, 1983). Propositions were numbered by the input cycle in which they were read. One variable scored the input cycle from the beginning of the entire passage, and one from the beginning of each paragraph.

Testing the Model

Method

Subjects. Sixty-four subjects from an undergraduate introductory psychology course at the University of Colorado participated in fulfillment of a course requirement.

Materials. Four texts were selected. They concerned wool production in Australia (94 propositions, from a junior high social studies text), the theory of continental drift (106 propositions, from a textbook on geology), San Francisco history (90 propositions, from a city guidebook), and breathing as a biological feedback mechanism (87 propositions, from a high school biology textbook.) After minor editing, each passage had three paragraphs and was 210 words long.

Procedure. The order in which subjects read and recalled the four texts was counterbalanced using a Latin-square design. Subjects received booklets with a practice text and the four stimuli texts, and were asked to read for comprehension, not to try to memorize verbatim. They read at their own pace. After they read each passage, they turned to the next page and wrote as much as they could remember in their own words.

Sixteen subjects wrote abbreviated protocols. If a subject gave abbreviated recalls for more than one text, falling into the bottom 20% of words recalled for that text, then another subject was run in that condition. Each recall protocol was propositionalized and compared with the original list of propositions from the stimuli passages. Analyses were performed on the number of subjects who recalled a given proposition.

Prowrtion recalled. Subjects wrote summaries that were on average about one quarter the length of the original texts. Subjects with at least three full protocols recalled 25.6 propositions per text on average, $\sigma = 11.7$. Propositions not relevant to the text were infrequent and excluded from the analyses.

The simplest passage, taken from the junior high school text, was best recalled. The effect of passage yielded $F(3,36) = 18.02$, $r = .27$, p c .0001, with means of 32.0, 25.5, 22.5, 22.4 propositions recalled for the texts on wool, continental drift, feedback and San Francisco, respectively. The effect of the order in which the texts were read was small and did not reach standard levels of significance, F(3,72) = 2.40, r = .04, p = .07.

Parameterization. The size of the working memory parameters -- the maximum number of propositions brought in during the input cycle, the maximum number at the end of the reduction cycle, and the reduction end fail-safe number -- were varied along with the strength parameter, a. Four working memory sizes were examined: large with bounds of 21, 11, and 5, moderate-large with bounds of 17, 9, and 4, moderate with 13, 7, and 3, and small with 9, 5, and 2 propositions for the input maximum, the reduction end maximum, and the reduction end minimum, respectively. The moderate size memory condition was based on a naive memory limit of seven propositions during processing. The network at the end of reduction was assumed to become a single chunk, allowing a maximum of six more propositions to be input on the next cycle. The reduction end minimum was set to maintain the base propositions in roughly a two to one range. The other memory bounds were set around the moderate limits, maintaining the same proportions.

Results

Two tests of the effects of varying the memory and the strength parameter were conducted. The first examined the ability of the model to determine reasonable clause boundaries at which to end each input cycle. The second examined the abiiity of the model to predict subject's recall. These tests suggest that a moderately large memory that favors recently input propositions may best approximate the reading process.

Using that set of memory parameter values, two additional tests were conducted. First, the residuals of the model were examined for each predicate category. The residuals were not significantly different across the different predicate classes. However, there was serial dependence in the errors. Second, the contents of working memory at the end of each cycle were examined for coherence. Coherence was maintained for 70% of the cycles. However, again the model appears prone to remain in error over several successive cycles.

Input clause boundaries. Three raters coded clause boundaries with an average of 98.8% inter-rater agreement. Disparities were resolved by discussion. The number of hits, correct rejections, misses and false alarms was tabulated, and d' (the distance in z-scores between the signal and the noise distributions, Green & Swets, 1966) was computed for each passage, memory size, and value of a. The results for this signal detection analysis appear in Table 4.

Insert Table 4 about here

The input phase was most successful with the two larger sets of memory parameters. Across all of the memory sizes, the majority of errors were due to false alarms. These were more common in the smaller memories, because there was a greater tendency to reach the maximum number of propositions that could be input into memory before reaching the end of a clause. With the memory parameters set to **17, 9,** and 4, the input memory bound was reached on only 2 of the 99 cycles (the continental drift passage went through 24 input-reduction processing cycles, the other three passages each underwent 25). With the moderately large memory boundaries about half of a sentence (*M* = 0.45) is processed **per cycle.**

As might be expected, because the strength parameter affects the input phase only indirectly via control in the reduction phase, changes in the value of a had little effect on input.

Predicting subjects' recall. Five independent variables were used in a multiple regression to predict the number of subjects who recalled a particular proposition. These independent variables were the number of entries in long-term memory, the average long term memory strength, the product of the two, the input cycle numbered from the beginning of the text, and numbered from the beginning of each paragraph. The adjusted-R%, all significant at the $p < .0001$ level, appear by passage for each of the various parameter levels appear in Table 5.

Insert Table 5 about here

. The simulation model was robust in predicting recall across the entire set of parameter values. On average, the model predicted 25.8 percent of the variance in recall, from a low of 17 percent to a high of 36 percent. For three of the four passages, the best performance occurred with the moderately large memory parameters.

The results for a varied by passage. The passages on wool production and continental drift did best across every memory size with the high a, the passage on biological feedback did so with the moderate a, and the passage on San Francisco did so with the low a. Across all passages, the high value of a, which stresses recency, was most predictive with large memory. With a large memory, older propositions are less likely to be forced out of memory by the memory bounds, and high values of a keep the contents of memory current via the ranking and rule application steps.

The best fits for the memory data overall come from the moderate large memory parameters and the high value of the strength parameter. With these parameters, the end of the reduction cycles were balanced between the upper memory boundary, the lower boundary, and between the two: Reduction stopped at the upper boundary 34 times, between the two boundaries on 36 cycles, and at (or in one case, due to the PRUNE action, one proposition below) the lower boundary on 29 cycles. These parameters were consequently used to calculate the significance of the regression weights for the five independent variables. Dummy variables were included to control for differences in the intercept across passages. The analysis for the average long-term memory strength yielded $t = 3.68$, $p < .0002$, the number of entries into long term memory store yielded $t = 6.29$, $p < .0001$, and the interaction between them, total strength, was not significant The input cycle parameters, numbered from the beginning of each paragraph and from the beginning of the passage as a whole yielded $t = -13.71$, p c .0001, and $t = -4.84$, $p < .0001$, respectively. The problem of multicolinearity between the first and second sets of variables was minimal.

Serial dependence in recall. Memory scores for the propositions of a text are, by definition, a time-series: a series of observations of the values that a variable takes ordered by time. Time-series analysis requires special statistical tools. Regression analyses of time series, for example, must be checked for violations of the independence assumption. Violations of independence typically lead to overstated significance levels. Such violations suggest that variables may be missing from the analysis.

The Durbin-Watson statistic is the most commonly used test for serial dependence in the errors of a regression. The statistic equals 2.0 when there is no serial dependence in the errors. The lower bound is 0.0, and the upper bound is 4.0. Low Durbin-Watson values indicate positive autocorrelation in the errors, that is, positive errors tend to follow positive errors, and negative errors tend to follow negative errors. High Durbin-Watson values indicate negative autocorrelation, that is, a tendency for the errors to oscillate between positive and negative values, from one to the next. Three sorts of values are defined for the statistic: those for which the series is accepted as having no dependence in the errors (i.e., values close to 2.0); those for which the hypothesis of independence is rejected (i.e., values closer to 0 or closer to 4); and those for which serial dependence or independence is indefinite.

The regression for two of the texts showed significant serial dependence in the errors. The passages on wool production and biological feedback showed independence, with Durbin-Watson values of 1.80 and 1.97, respectively. The continental drift and San Francisco passages, in contrast, both showed significant positive autocorrelation, with Durbin-Watson values of 1.26 and 1.19, respectively, both $\underline{p}s < .01$. These results suggest that when the model performs poorly in the latter two passages, it tends to do so for several successive cycles.

All propositions across the four passages with extreme residuals were examined to explore the autocorrelation problem. In the continental drift passage, a clause midway through the second paragraph mentioned that all continents were once part of a single landmass. The remainder of the text concerned this key notion, and this sequence of propositions was memorable, far more so than predicted by the model. It was one of the few phrases written in what would colloquially be called plain English. In the San Francisco passage there were two sequences of propositions of extreme residuals of the same sign. Both sequences were directly related to San Francisco itself (i.e., the size of the Bay, and the city’s previous name), rather than the surrounding, now largely uninhabited, area. The title of the passage and the subjects’ previous knowledge would both be expected to have made the former information relatively more salient

Recall and predicate class. The memory data were also used to investigate the appropriateness of the various ranking and reduction rules associated with each predicate class. Table 6 shows the mean and the variance of the residuals derived from the simulation using the moderate-large memory for each of the three levels of a. Positive residuals indicate that more subjects recalled a particular type of predicate than the model predicted. Either the model engages in less processing of the propositions in that category than is necessary, or it held *these* propositions in memory for too few processing cycle. Negative residuals indicate the reverse: too little processing or too rapid elimination from memory. The "no-model" figures provide a baseline for comparison. These figures are calculated controlling only for the average recall in each of the stories from which the individual propositions appeared. Positive and negative means indicate whether propositions with a given predicate class are more or less memorable that average.

Table 6 also shows the standard deviations of the residuals. Standard deviations are included because an inappropriate predicate class reduction rule can also degrade the overall performance of the model by increasing the noise in the model's predictions. A large standard deviation of the residuals indicates diversity among the members of the category. This can occur either because a rule has been misspecified or because two or more categories have been treated as one.

Insert Table 6 about here

All of the predicate classes in the immediate resolution category showed negative residuals in the no model column, an indication of their relatively poor recall across subjects. With a = .7, the model accounts for this lack of recall, without greatly increasing the standard deviation of its errors in prediction. The initial small standard deviation for the no model immediate resolution propositions is attributable to a floor effect.

Before the predication semantics model was applied, the descriptive predicates, as a group, received average memory scores. Of the six better remembered predicate categories, three were specific quantifiers, and three were in the identification category. The model reduced both the mean and the standard deviation in the former case. Examining all instances of identification, it appears that NAME, SET and IDENTITY are often focal, in that they give new information, whereas the remaining identifiers are typically incidental. These latter results suggest that the identification category may need to be subdivided either by adding a new reduction order category or by changing the specific reduction rules to induce additional processing.

As expected, most of the coordination predicate propositions were more memorable than the average proposition. The model improved the standard deviation of the residuals for all four coordination predicate categories. Only one of the categories, contrasts, showed a substantial mean error. The model reduced this error and reduced the standard deviation of the residuals for all four of the coordination predicate categories.

As predicted, nonclassified predicates were remembered at better than average levels. Nonclassified predicates accrue long term memory strength by lasting through several cycles of processing, which high values of a inhibit, The nonclassified predicates had smaller residuals at lower values of a than at higher values.

To test whether the differences between the mean residuals of the various predicate classes were significant, that is, whether the model accounted for some categories better than others, an one-way ANOVA was run, with the negation category removed (it had only one instance). The results for the a = .7 model yielded *F*(9, 366) = 2.02, *p* c .04. A Duncan's multiple range test revealed no specific significant differences between the various groups. Without the nonclassified predicates in the analysis, the results of the analysis were *F*(8, 256) = 1.30, *p* > .2.

Strength: Long term versus working memory calculations. One minor improvement was possible, due to the manner in which strength is calculated. Recall that a proposition is held to have a connection when it uses a new proposition as an argument, but not when it uses an old proposition. This was done to direct the processing of older propositions from the central node of the network. It does, however, insufficiently represent the potential long term memory strength of such propositions. Propositions that used only other propositions as arguments had significantly higher residuals (*n* = 71, *M* = 2.5) than did those without (*n* = 306, *M* = -0.6), *t* = 2.32, *p* < .03. *Coherence.* The end of cycle propositions were rated as to whether they were coherent representations of the text. Using a conservative criterion of no errors, coherence was maintained on 72.7 percent of the trials. Two texts showed tendencies to remain incoherent over successive cycles. The joint autocorrelation of coherent and incoherent cycles for the drift and feedback texts combined was *r* = .24, *p* <.05 (one-tailed), whereas the joint autocorrelation for the wool production and biological feedback texts was *r* = .02. When the propositions in working memory become incoherent, they tend to remain problematic for several cycles in a row. The texts that show serial dependence in incoherence were those that show autocorrelation in the residuals of the memory scores.

The problem of serial dependence in both memory predictions and in coherence were not unrelated. The coherence problem would appear to occur because the simulation does not, unlike real readers, reread when necessary. Cycles in which the highly memorable strings of propositions were present were not incoherent. However, in these cases, sets of cycles before these propositions were read into working memory and sets of cycles after these propositions left working memory did show degraded levels of coherence.

Appendix B shows the processing by the simulation of the first paragraph of the Continental Drift passage to illustrate how the model maintains coherence.

Discussion

The results of the experiment suggest that the predication semantics model is robust The simulation worked well over an array of memory size and strength parameters. Optimal input was found with a moderately large memory size, which read about half of a sentence per processing cycle. Most clause boundaries were then identified correctly, and input performance did not improve by using a larger memory span. The same memory parameters were optimal in predicting subjects' recall. With the exception of unclassified predicates, larger memory spans worked well with a high strength parameter, which favors recent over older propositions in working memory. With these parameters, there was no significant information remaining among the predicate categories. These results suggest that the model emulates a primary processes in text comprehension and recall. Predicate categories provide semantic information that helps to initiate and control automatic processes in reading that require little background knowledge.

The simulation of the model maintained coherence in working memory on the majority of trials. This represents strong evidence in favor of the importance of predicate semantic categories in guiding reading and summarization processes. Two minor problems were found that account for most cases in which the contents of working memory became at all incoherent. Fit, there were instances in which the model stopped the reduction phase with all but one of the propositions held as part of a highly intercormected network. This reflects the weakness of relying solely on the average as the oniy strength measure used to end the reduction phase. A kurtosis measure might be. helpful as well. It seems reasonable to expect that an active highly interconnected network inhibits the intrusion of a low strength node, rather than maintaining it Second, there were instances in which the model simply lost track of gist, because a central concept was dropped, did not appear until the next cycle, or was indicated by a superordinate node, such as the title of the passage. This problem can be remedied by amending the simulation so that it can use the contents of long-term memory to indicate initial memory strength. Neither of these problems presents a severe challenge to the model.

In the current version of the simulation, transfers from long term memory only occur when a proposition makes an explicit reference to another proposition that is no longer in working memory. With human readers, information flows more frequently between long term and working memory. This problem with the current simulation may be remedied simply by allowing it to reread when the average strength of the network of propositions in working memory becomes too low given the number of propositions remaining. The simulation did poorly when concepts introduced later in a passage were needed to identify which information was central to gist from an earlier section. Authors sometimes structure the initial part of passages such that a fan of information is given about a central topic, thereby highlighting the overall importance of the topic, before developing one particular area in the later body of the text. The ability to reread or recall from long term memory, once the focal area is highlighted, then becomes crucial to bring back what can now be identified as central information. Similarly, historical texts are often told using the original time course of events. Current history, likely to be most familiar to the reader, then comes in the latter part of the text

The predication semantic model builds on previous case grammar approaches to text comprehension. We must note a particular debt to the Kintsch and van Dijk (1978) model, especially to the notion of macrorules that produce macropropositions. They briefly sketched several general rules, such as removing propositions that do not aid in the interpretation of subsequent propositions and generalizing sets of propositions into one denoting a superordinate idea. These ideas about deletion and generalization are incorporated in the manner in which strength calculations are made and integrated into the ranking stage of the predication semantics model. We have gone several stages further. Their model did not adopt intermediate-level reduction rules based on predicate class. It was based on the more rigid hierarchical tree notion, rather than using a propositional network. Most importantly, it did not maintain coherence. However, the goal of extracting the gist of the text within the constraints of memory resources is clearly the same as that of our reduction processes.

Improving the model

In trying to make our processing rules well defined, certain problems have become clear. One problem is that the model does not give sufficient processing resources to unclassified predicates, which are primarily verbs, if a high strength value is used. High strength values otherwise gave the best overall levels of processing. There are two alternative solutions. The first solution is to set two different strength criteria for bringing propositions from the upper reduction categories into the second reduction category, a higher one for description and coordination predicate propositions, and a lower one for unclassified predicates. The problem with this solution is that it is likely to degrade the coherence of the propositions in working memory at the end of the processing cycle. The second solution is to define a set of reduction rules for the unclassified verbs. Some initial progress can be made by creating a separate predicate category of verbs of movement, which in our sampling of these texts and others tends to cover about a third of the unclassified predicates.[4] Also, the information that verbs supply in tense markers needs to be incorporated.

A second problem is that when the simulation of the model goes wrong, it stays wrong for several successive intervals, both in terms of predicting summarization and recall and in terms of coherence at the end of a reduction cycle. Part of this problem is that the simulation does not incorporate a sufficiently complex memory. Higher prior strength values ought to be assigned for what readers are likely to preconceive as the central topics (Spilisch, Vesonder, Chiesi, & Voss, 1979). Arguments that appear later in a text ought to be able to increase the strength of earlier propositions that have left working memory, simulating the backward propagation induced by rethinking or rereading. Concepts with higher strength value ought to be searched first, given that they are more rapidly reinstated (e.g., Lesgold, Roth, & Curtis, 1979). It ought to be possible to reinstate networks of propositions, such as a schema, script or plan, such as that which occurs when readers encounter a topic about which they have a large amount of prior knowledge (Britton & Tesser, 1983). The inferences often required to fill gaps, when they occur, require such processes, and the absence of such inferences within the current simulation degrades performance, just as it degrades the mental representations of readers who fail to do so as well (Britton & Gulgoz, 1991; Britton, Van Dusen, Glynn & Hemphill, 1990).

Creating a more realistic long term memory structure would have drawn from the central focus of the current work, but several features of the simulation make the inclusion of more memory processing resources relatively easy. First, the reduction process in the model frees resources by decreasing the storage demands. These resources can then be allocated to inference construction, or used to predict when inference construction is likely to occur. Second, the simulation currently includes a procedure for focusing the inference construction process, via its strength calculation. One would expect that those concepts with the greatest strength would be those most likely to be the focus of additional inferential processing. Third, there is no requirement that the strength calculations used in working memory be the same as those used in long term memory. The simulation was primarily constructed to maintain coherence, and accurate prediction of memorability was subjugated to this end. Location of the central node of an older network clearly calls for different strength calculations between sequencing propositions for reduction, and defining their long term memory strengths.

One final issue is the need for more sophisticated reduction rules. For example, the current model handles the reduction of negation by deleting the entire set of propositions negated. We have not developed interactive rules that would otherwise prevent changes in meaning. Undoubtedly such rules would improve the predictions of the model. The success of the rules even at the current restricted level of detail demonstrates the importance of the semantic information found in predicate categories.

References

Anderson, J. R. (1983). *Architecture of Cognition*. Cambridge, MA: Harvard University Press.

Baddeley, A. D., & Hitch, G. (1974). Working memory. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 8). New York: Academic Press.

de Beaugrande, R. (1980). *Text, Discourse, and Process*. Hillsdale, NJ: Earlbaum.

Britton, B. K., & Guigoz, S. (1991). Using Kintsch's computational model to improve instructional text: Effects of repairing inference calls on recall and cognitive structures. *Journal of Educational Psychology, 83*, 329-345.

Britton, B. K., & Tesser, A. (1983). Effects of prior knowledge on use of cognitive capacity in three complex cognitive tasks. *Journal of Verbal Learning and Verbal Behavior*, *26*, 421-436.

Britton, B. K., Van Dusen, L., Glynn, S. M., & Hemphill, D. (1990). The impact of inferences on instructional text. In A. C. Graesser & G. H. Bower (Eds.) *The psychology of learning and motivation.* (Vol. 25, pp. 53-70). New York: Academic Press.

van Dijk, T. A., & Kintsch, W. (1983). *Strategies of Discourse Comprehension.* New York: Academic Press.

Fiimore, C. J. (1968). The case for case. In E. Bach and R. T. Harms (Eds.) *Universals of Linguistic Theory*. New York: Holt, Rinehart & Winston.

Fodor, J. A., Bever, T. G., & Garrett, M. F. (1974). *The Psychology of Language*. New York: McGraw-Hill.

Green, D. M., & Swets, J. A. (1966). *Signal detection theorv and psychophysics*. New York: Wiley.

Grice, H. P. (1975). Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and semantics: Vol. 3. Speech acts* (pp. 64-74). New York: Academic Press.

Gruber, J. (1965). *Studies in lexical relations.* Doctoral dissertation, M.I.T., Cambridge, MA.

Haviland, S. E., & Clark, H. H. (1974). What's new? Acquiring new information as a process in comprehension. *Journal of Verbal Learning and Verbal Behavior, 13*, 512-521

Jackendoff, R. (1972). *Semantic Interpretation in Generative Grammar*. Cambridge, MA: MIT Press.

(Eds). Understanding expository text (pp. 89-107). Hillsdale, NJ: Erlbaum.

Kieras, D. E. (1989). An advanced computerized aid for the writing of comprehensible technical documents. In B. Britton & S. Glynn (Eds.) Computer writing environments: Theory, research and design (pp. 143-168). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kintsch, W. (1988) The role of knowledge in discourse comprehension: A construction-integration model. Psychological Review, 95, 163-182.

Kintsch, W., & van Dijk, T. A. (1978). Toward a model of text comprehension and production. Psychological Review, 85, 363-394.

Kintsch. W., & Keenan, J. M. (1973). Reading rate and retention as a function of the number of propositions in the base structure of sentences. Cognitive Psychology, 5, 257-274.

Lesgoid, A. M., Roth, S. F., & Curtis, M. E. (1979). Foreground effects in discourse comprehension. Journal of Verbal Learning and Verbal Behavior, 18, 291-309.

Mandler, J. M. (1978). A code in the node: The use of the story schema in retrieval. Discourse Processes, 1, 14-35.

Mandler, J. M. & Johnson, N. S. (1977). Remembrance of things parsed: Story structure and recall. Cognitive Psychology, 9, 111-151.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on capacity for processing information. Psychological Review, 63, 81-97.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). Five papers on WordNet. CSL Report 43, Princeton, NJ: Cognitive Science Laboratory.

Miller, J. R., & Kintsch, W. (1980). Readability and recall of short prose passages: A theoretical analysis. Journal of Experimental Psychology: Human Learning and Memory, 6, 335-354.

Norman, D. A., & Bobrow, D. G. (1975). On data-limited and resource-limited processes. Cognitive psychology, 7, 44-64.

Perrig, W. & Kintsch, W. (1985). Propositional and situational representations of text. Journal of Memory and

Language, 24, 503-518.

Pratkanis, A. R., Greenwald, A. G., Leippe, M. R., & Baumgardner, M. H. (1988). In search of reliable persuasion effects: III. The sleeper effect is dead. Long live the sleeper effect. Journal of Personality and Social Psychology, 54, 268-273.

Schank, R. C. (1980). Language and memory. Cognitive Science, 4, 243-284.

Schmalhofer, F. J. (1982). Comprehension of a technical text as a function of expertise. Doctoral dissertation, University of Colorado.

Spilish, G. J., Vesonder, G. T., Chiesi, H. L., & Voss, J. F. (1979). Text processing of domain-related information for individuals with high and low domain knowledge. Journal of Verbal Learning and Verbal Behavior, 18, 275-290.

Trabasso, T. & van der Broek, P. (1985). Causal thinking and the representation of narrative events. Journal of Memory and Language, 24, 612-630.

Turner, A. A. (1987). The propositional analysis system, Version 1.00. (Tech Rep. No. 87-02). Institute of Cognitive Science, University of Colorado.

Turner, A. A. & Greene, E. (1978). Construction and use of a propositional text base. JSAS Catalogue of Selected Documents in Psychology, MS 1713.

Winograd, T. (1983). Language as a cognitive process: Volume 1: Syntax. Reading, MA: Addison-Wesley.

Appendix A: Reduction Rules

Description medicates. The rules for the description predicates appear in short form in the right column of Table 2. For modification predicates and auxiliary verb predicates, the CORE action alone was used. The modified verb, noun, and non-auxiliary verb were always selected as the core argument, because modifiers and auxiliaries lose meaning without the arguments they modify as anchors, whereas the reverse does not hold. The success of these rules clearly depends on the other steps of a reduction phase, given that these rules can change the meaning of a set of propositions. For example, “he should have called” becomes “he called.” But, as was the case with PART predicate class propositions, if his calling or not is the primary issue, usually shared arguments initially will protect the MODAL proposition from untimely reduction, and later slate both it and the verb proposition it references for reduction on the same reduction cycle,

The rules for the next set of description predicates are only slightly more complex. For the six predicate classes in the identification category, both arguments can generally stand alone. Consequently, the rules for the identification predicates followed the reasoning outlined above with PART. The member was used as core rather than the set, the possession rather than the owner, the object rather than its material, and the instance rather than the identity, but only when the latter arguments could not themselves be reduced. These choices are based on how identification propositions are themselves likely to be embedded as arguments and the context in which they are likely to rest, For example, choosing the member as the core rather than the set may not at first seem generally correct, given that a set may be used to describe a member, or a member used to illustrate a set. However, the former usage is common when a SET proposition is embedded in another proposition as an agent, patient, or object. Otherwise the set argument is embedded directly. Similarly, context generally determines that the owner in a POSSESS proposition is less likely to be the focal topic than the related object. Agents commonly operate on their own possessions, so the owner of a possession can often be inferred from other propositions in memory. Finally, COMPOSITION and IDENTITY propositions often serve a role similar to QUALIFY propositions, such that the latter arguments stand poorly on their own. In every case, the initial REDUCE action helps guard against exceptions.

REDUCE was not used with the NAME predicate class rule because the name argument is generally not a proposition. And unlike those for the other identification predicate propositions, the rule for NAME propositions does not select a predetermined argument as the core. Rather it selects the BEST argument as the core as determined by the context. CORE BEST takes a stronger over a weaker argument. This requires that both arguments are propositions and their strength values differ. Failing this, the argument with more connections is selected. If there are an equal number of connections, then a proposition nested as an argument is taken over a simple argument. Only to avoid a tie does the rule then "favor" a predetermined argument as the core. The NAME rule used CORE BEST rather than CORE with a default argument, because the proposition frequently serves both to introduce a name and to replace an old name with a new. CORE BEST lets the context decide.

The rules for the four predicate classes in the auantifkation category transform specific information into more general information. QUANTIFY predicate propositions, the most general in the category, quantify with an imprecise word or phase (e.g., "abundant," "scarce," "adequate"). The rule for QUANTIFY propositions, as with the majority of predicate classes in the identification category rules above, uses REDUCE and then CORE. However, both actions focus on the quantified object. Simple arguments rather than embedded propositions usually provide quantification. Consequently, applying the REDUCE action to the quantifying argument generally would not distinguish important from unimportant propositions. NUMBER and RANGE predicate propositions, which provide intermediate specificity, quantify with numbers. NUMBER and RANGE propositions are reduced by the TRANSFORM action to become QUANTIFY propositions. The explicit numeric information is replaced with a more general verbal description. Real world knowledge is required to determine whether a particular numeric argument constitutes many, much, some, little, or few, as the case may be. The computer simulation leaves these arguments indefinite. Finally, EXTENT propositions, the most specific class of quantification descriptors, quantify by embedding propositions of the three predicate classes above (e.g.. "500 to 600 meters of rope" is a RANGE proposition about meters embedded in an EXTENT proposition about rope.) The reduction rule for EXTENT propositions with embedded QUANTIFY propositions uses the TRANSFORM action to change them to QUANTIFY propositions. Embedded NUMBER and RANGE propositions are first reduced to QUANTIFY propositions. The quantifying word from the old embedded QUANTIFY proposition is applied to the quantified object in the EXTENT proposition in the new EXTENT proposition. For example, "500 meters of rope" would become "many meters of rope," and then "much rope." These rules avoid applying the quantifying argument in a NUMBER or RANGE proposition directly onto the quantified object in the embedding EXTENT proposition, such that "500 feet of rope" does not become either "many feet" or "500 ropes." The rules also allow for the fact that in concise form, precise numeric arguments are often restated or recalled as approximate verbal summaries.

The rule for the negate category was discussed above. **REDUCE is** applied to the embedded argument, until it cannot be reduced further. If there can be no reduction, and if the embedded proposition is not protected, then the **PRUNE** action is applied. Recall that no proposition may be deleted if referenced by another. The **PRUNE** action can cause the number of propositions at the end of the reduction cycle to fall below the lower memory boundary.

Coordination predicates. Rules for coordination predicate classes appear in the right column of Table 3. Both predicate classes in the communication category use the **CORE** action. The rule for COMMUNICATION propositions selects the message, and drops the speaker and recipient. The recipient is generally treated as unimportant, and never appears in many types of text. Dropping the speaker requires more explanation. The rule could be constructed to give more weight to the speaker by the use of the **REDUCE** action on the message before the **CORE** action. Repeated reduction would strengthen the long-term memory connection between the speaker and the communication. The **CORE** action is taken immediately because information necessary to the text is often attributed to an authority (e.g., "Newton argued . . .") when such information can stand alone. These attributions serve only for emphasis, as background unimportant to gist. Moreover, idiosyncratic messages can be inferred back to the speaker, making the communicator, but not the communication, redundant_ Similarly, it is common in narrative fiction that when relaying dyadic conversations, the speakers are no longer named after the initial exchange, such that the reader must rely on turn-taking cues for clarification. Finally, social psychological research suggests that information content of messages can become disassociated from the communicator even in cases in which the reliability or motivation of the communicator would raise questions about the veracity of what was said (Pratkanis, Greenwald, Leippe, & Baumgardner, 1988). Undoubtedly, dropping the speaker is more likely to lead to a confusing set of propositions in working memory than is dropping the recipient. The simulation currently does not check whether there are references to other speakers in long term or working memory, which would suggest that the speaker information must be retained while the message is processed into gist.

The rule for TOPIC predicate propositions took the BEST argument, rather than a default. Either the header or the topic (e.g., “Newton’s theory” or “planetary motion”) may best serve in developing the gist of the text, and the context determines the choice.

The rules for the temporal and locative category of predicates use several stages of reduction. These strengthen the ties between the object or event and the time or location argument in long-term memory. These ties are important because time and location information plays a superordinate position in organizing the sequence and placement of events and objects, links stressed by exposure to academic tests. The strategy in the reduction rules for propositions with LOCATION and TIME predicates is the same. If a proposition is embedded within a proposition of the same predicate class, the rule adopts the strategy of changing specific information into more general information. The embedding is removed by taking the embedded object or event as the core (e.g., “he parked in the lot by the bank” becomes “he parked by the bank.“) The rules then attempts, in turn, to REDUCE the former argument, to REDUCE the latter, before finally using the CORE action on the former. There are texts in which the temporal or locative argument is more central to gist (e.g., “she attended a party at Buckingham Palace”). However, the ranking procedure again helps prevent untoward reduction. TIME and LOCATION propositions both retain whatever preposition appeared in the text as part of the predicate.

The rules for contrast predicates, like those in the quantification category, make specific information more general. NUMERIC-CONTRAST propositions compare the number of members in two sets, a comparison and a standard The original predicate that defined the relationship between the two is used as part of the predicate. The rule uses TRANSFORM to change the predicate class to QUANTIFY, adapts the original contrast predicate to provide the sense for the quantifying argument, and applies the PRUNE action to the standard set (e.g., “there are more Mercedes. than Rolls” becomes “there are many Mercedes.“) QUALITATIVE-CONTRAST compare two sets along a given dimension. The direction of the relationship is included as part of the predicate. The rule, following a parallel logic, uses TRANSFORM to change the predicate class to QUALIFY, adapts the dimension entailed in the original proposition as the modifier in the revised QUALIFY proposition, and applies PRUNE to the standard (e.g., "Mercedes go faster than Rolls" becomes "Mercedes go fast"). Finally, EQUALITY-CONTRAST propositions use the CORE BEST action, following the reasoning used for TOPIC propositions.

Propositions in the logical relations category are those most likely to provide superordinate structure to the text, often containing arguments derived from different sentences in the text Consequently, the reduction rules used with logical relation propositions are oriented toward building a base for propositions read on upcoming input cycles. The rules for all of the predicate classes in this category use the PROTECT condition, affected by the strength of the proposition and by the value of a. The PROTECT condition, when in effect, increases the likelihood that all arguments are identified before active changes are invoked.

The rules for CAUSE and CONDITION then use CORE BEST, rather than selecting one as the default, because either argument may serve as the central condition in later propositions. The cause may have additional effects or the effect may initiate a second. CORE is used rather than REDUCE to provide an intact link for subsequent connections in the causal chain (Trabasso and van der Brook, 1985).

The rule for propositions with PURPOSE predicates takes a somewhat different tack: These propositions are used to bind both agents' actions and goals (e.g., "he borrowed the car to get home"), and agents and affordances (e.g., "he was born to lead"), covering the Aristotelian notion of causa ut, that which entails a desired effect, and of causa materia, that which holds a natural inclination toward some use or end, respectively. With either manner of PURPOSE proposition, the arguments typically lose meaning when separated from each other or from the predicate, especially in the case of affordances. Agents and the associated goals or purposes generally serve jointly as starting points in causal sequences, as opposed to links or branches along the chain (cf. Schank, 1980). As such, although the rule for PURPOSE favors the second argument, as do CAUSE and CONDITION, the PURPOSE rule begins active reduction with REDUCE on the first, then the second arguments, which strengthens their relationship in both working and long-term memory.

The rule for CONCESSION uses CORE on the concession argument and PRUNE on the original assertion. The latter action is done concurrently with the first given that the nature of a CONCESSION predicate is to render a change in the truth value of the initial argument. The PRUNE action may be more parsimonious than most readers, but it is necessary because the CORE action removes the predicate.

CONJUNCTION predicates join the arguments of both simple lists and multiple clauses, and the rule covers both two and three argument cases. (Beyond three arguments, CONJUNCTION propositions are embedded within other CONJUNCTION propositions.) In the three argument case, when the PROTECT condition is not in effect, the third argument is dropped from the proposition, although not from working memory. In a multiple clause proposition, especially with complex clauses, the early arguments can become unimportant while the final argument is still the active focus of the text. By dropping the third argument from the proposition, the earlier and now less relevant arguments can be reduced without reducing the argument that has become the primary concern. If they are to stay important, they must do so on their own merits. In addition, in multiple argument simple lists, owing to the maxims of good communication (Grice, 1975), the first, not the last, argument typically makes the best single argument header for the list as a whole. Dropping a third argument generally reduces a simple list toward the most representative item. CORE BEST is used in the two argument case. Either argument may serve as the focus in upcoming propositions with multiple clause CONJUNCTION propositions, and taking the BEST argument resolves the problem of the first argument having lost relevancy.

Unclassified predicates. Propositions with unclassified predicates are almost exclusively those based on action verbs. These propositions are deleted after propositions embedded in them, or in which they are embedded, have been reduced. In the rare cases in which one unclassified proposition is embedded in another, the rules delete the higher level proposition first.

Special cases. Some parts of rules needed for special cases do not appear in the tables. First, the model allows for the reduction of time spans, physical distances, and numeric differences in TIME, LOCATION and NUMERIC-CONTRAST propositions, respectively. Second, SET, COMMUNICATION, and CONJUNCTION predicate classes can take specific predicates that alter their usual course of action, for example, the SET rule treats "one of the" differently from "any other than"; the CONJUNCTION rule treats an "and" differently from an "either/or." Third, the IDENTITY rule checks for shared arguments embedded within its arguments. This checking has effects that prevent phrases like “a red rose is my favorite rose” from becoming tautologies. Finally, there are special rules initiated to resolve cases in which the model would otherwise stop processing before reaching the end of the text. For example, these rules ensure that at least one proposition is input on each cycle. Similarly, there is a procedure that ensures that two propositions do not repeatedly take turns trying to reduce one another.

Appendix B

To show how the model maintains coherence, the processing of the first paragraph of the Continental Drift text using the memory parameters of 17, 9 and 4 propositions, and an α of .7 is presented. The original passage is:

> "Wegener proposed a theory of continental drift when geologists were beginning to find conventional theories of continental permanence inadequate. Previous continental drift theories held that some catastrophic event initiated continental displacement. In contrast, Wegener proposed that the same forces that produce great folded mountain ranges, displaced the continents. He presented evidence from such a range of sciences that his theory could not easily be ignored."

Cycle 1. Three propositions are input. No reduction is necessary because there are fewer than ten propositions and the average strength exceeds the criterion of 84.5 percent, halfway between that of a new proposition with one connection (i.e., $S_{0,1} = 0.91 = 0.7 + 0.7 * (1 - 0.7)$) and such a proposition on the next cycle again with a single connection (i.e., for $S_p = 0.91$, $S_{m,0} = 0.27 = 0.91 * (1 - 0.7)$, so $S_{m,1} = 0.78 = 0.27 + 0.7 * (1 - 0.27)$).

Cycle 2. The second cycle brings the remaining six propositions from the first sentence. No reduction occurs for the same reasons as in the first cycle.

Cycle 3. The third input cycle reads another six propositions before ending input. The contents of working memory are then:

P1 *PROPOSE* [wegener, P2]	92
P2 TOPIC:of [theory, P3]	97
P3 *DRIFT* [continents]	97
P4 TIME:when [P1, P5]	93
P5 PROCESS:begin [geologists, P6]	78
P6 FIND [geologists, P7, inadequate]	78
P7 QUALIFY [P8, conventional]	78
P8 TOPIC:of [theory, P9]	97
P9 *PERMANENT* [continents]	97
P10 TIME:previous [P12, P1]	99
P11 *DRIFT* [continents]	99
P12 TOPIC:of [theory, P11]	99
P13 *HOLD* [P10, ?]	91
P14 QUALIFY [P15, some]	91
P15 *OCCUR* [catastrophe]	91

Unclassified predicates appear in italic capitals. Approximate strength values appear to the right of the propositions in percent. A "?" as an argument, as in P13, indicates that the input cycle ends before the argument is activated.

With 15 propositions in working memory, at least six must be deleted by reduction.

The proposition ranked first is P3, which is redundant with P11. It is not reduced. P3 is an unclassified predicate, and consequently can only be reduced by deletion. However, if P3 were deleted then P2 would have a blank argument, which is not allowed to occur. For the same reason, P11 cannot yet be reduced because it is embedded in P12.

With no reducible propositions in the first reduction category, and none currently in the second, the simulation moves to the third category, the description predicates. There are three: P5, P7, and P14. As the weakest and oldest, P5 is reduced first. PROCESS predicates are reduced by the CORE action. The action replaces P5 as the second argument of P4 with P6, and deletes P5 from working memory. P7 and P14 are both QUALIFY propositions, which are also reduced by the CORE action. "Conventional theory" becomes "theory" and "some catastrophe occurred" becomes "a catastrophe occurred."

With the deletion of P14, P15 becomes unconnected, its strength falls to .7, and it becomes the only member of the second reduction category. The proposition, having an unclassified predicate, is reduced by deletion.

At this point, P6 is the weakest proposition, and it has no connections to a new proposition. However, it still retains P8 as an argument. The simulation therefore reduces P8, a TOPIC proposition. The rule calls for the CORE BEST action. The two arguments of the proposition are "theory," which has two connections (i.e., to P2 and P12), and P9, which has one connection (i.e., to P9 itself). As such, the first argument is selected by the BEST action as the core. P6 becomes, in effect, "geologists found theories inadequate."

With ten propositions left in working memory, at least one more must be reduced from memory. P9 is the weakest predicate. It is an unclassified predicate, but is not protected because it has no active connections. Since it is not an argument in any other proposition, it is simply deleted. There ate now few enough propositions with sufficient average strength to end the reduction phase. The propositions in working memory read,

> "Wegener proposed a theory of drifting continents when geologists found theories inadequate. Previously, theories of drifting continents held . . . "

<u>Cycle 4.</u> The fourth input cycle reads to the end of the second sentence. When P16 is read, the second

argument of P13 is filled, and P15 is recalled from long term memory. Had previous reduction cycles changed P15, the last version of the proposition before it was previously deleted from working memory would be the one recalled. The contents of working memory now read:

P1 PROPOSE [wegener, P2]	93
P2 TOPIC:of [theory, P3]	78
P3 DRIFT [continents]	97
P4 TIME:when [P1, P6]	27
P6 FIND [geologists, theory, inadequate]	78
P10 TIME:previous [P12, P1]	79
P11 DRIFT [continents]	98
P12 TOPIC:of [theory, P11]	79
P13 HOLD [P10, P16]	78
P16 CAUSE [P15, P17]	99
P15 OCCUR [catastrophe]	76
P17 DISPLACE [continents]	99

The proposition ranked first for reduction is P4, which falls into the second reduction category. The rule for TIME predicate propositions first attempts to reduce P6, which is nested within P4 as the temporal marker for P1, but cannot because P6 is protected by its strength, which places it in the fifth reduction category. The TIME rule then attempts to reduce PI, but cannot for the same reason. The TIME rule, having failed to be able to satisfy the first three conditions, then applies the CORE action. Since P4 is not used as an argument in any other proposition, the CORE action simply acts to delete the P4 from working memory.

Once P4 is deleted, P6 loses its only connection, and falls into the second reduction category. It is deleted. The CORE BEST action is then applied to P2, the weakest member of the third reduction category. There are then nine propositions in working memory, with sufficient average strength to end the reduction phase. The contents of working memory yield:

> "Wegener proposed drifting continents when previous theories of drifting continents held that a catastrophe caused continental displacement"

Cycle 5. the fifth reduction cycle, three more propositions are read into working memory, P18 to P20. (Only propositions from the earlier cycle are affected, so we do not list the contents of working memory here.) The program first tries to reduce P3, which is redundant with P11, but cannot because P3 is embedded in P1, which is still protected by having an active connection with P19. P11 is ranked second. P11 is embedded in P12, which is not protected. P12 is a TOPIC predicate, which calls for the CORE BEST action. P12 has two arguments, “theory” and P11. The former has no connections, so P11 is selected as the core argument P12 appears as an argument in P10, so the core action replaces P12 with Pll in P10, and deletes P12.

P3 and P11 again are ranked first and second, and again P3 cannot be reduced. The program tries to reduce P11, which is now embedded in P10, which in turn becomes the candidate for reduction. P10 marks Pll as having occurred previous to P1. Neither P1 or Pll can be reduced, so the TIME proposition rule selects P11 as the core. P10 is embedded in P13, so P11 becomes the first argument in P13, and P10 is deleted.

P3 is again ranked first, but now Pl can be reduced because it is no longer used as an argument in another proposition. Deleting it does not create an empty argument. Once it is deleted, P3 falls into the second reduction category and is also deleted. There are then eight propositions in working memory with sufficient strength to end the cycle.

Cycle 6: Contents of working memory at the beginning of the sixth reduction cycle are:

Pll DRIFT [continents]	76
P13 HOLD {P11, P16}	76
P16 CAUSE [P15, P17]	76
P15 OCCUR [catastrophe]	76
P17 DISPLACE [continents]	76
P18 CONTRAST:in-contrast [P19, P13]	28
P19 PROPOSE [wegener, P20]	78
P20 ECONTRAST:same [P21, ?]	93
P21 CAUSE [forces, P23]	97
P22 INTENSIFY [P24, great]	97
P23 POSSESS [P22, folds]	97
P24 QUALIFY:of [ranges, mountains]	91

Only one proposition, P22, falls into the first reduction category. P24 is the core argument, which replaces P22 as the first argument in P23. P22 is deleted. P18 is a generic contrast, in which a comparison set is defined as different from a standard set, without defining the dimension on which the two differ. The reduction rule applies the CORE action, keeping the comparison set. The rule then applies the PRUNE action to, in succession, P13, P11, P16, P15, P17, P18 and P19. Only four propositions remain, the minimum allowed, thus ending the cycle. (Average strength exceeded the criterion level, but with four propositions, this was not necessary.)

Cycle 7. The seventh input phase reads one proposition, which ends the third sentence. With only live well connected propositions in working memory, no reduction occurs. The contents of working memory read:

> “The same forces that caused mountain ranges to have folds displaced continents.”

Cycle 8. The eighth cycle reads in all but the final two propositions of the paragraph. The contents of working memory at the beginning of the reduction cycle are:

P20	ECONTRAST:same [P21, P25]	22
P21	CAUSE [forces, P23]	78
P23	POSSESS [P24, folds]	76
P24	COMPOSITION [ranges, mountains]	76
P25	DISPLACE [forces, continents]	78
P26	PRESENT [wegener, P27]	98
P27	DERIVE-FROM [evidence, P29]	97
P28	CAUSE [P26, P32]	97
P29	QUANTIFY (sciences, many]	90
P30	POSSESS [wegener, theory]	90
P31	MODAL:could [geologists, ?]	90
P32	NEGATE [P31]	97

The CORE action deletes P20, which is not referenced by another proposition, and a process of linear deletion occurs again. When P20 is deleted, the strength values of P21 and P25 fall. The reduction rule for P21, a CAUSE proposition, applies the CORE BEST action, which deletes it. P23 then falls into the second reduction category as the weakest member. The reduction rule for POSSESS propositions deflects action to P24. When the CORE action is applied to P24, P23 becomes weak and unconnected, and is deleted from memory. With this last action, there are eight propositions in working memory, with sufficient strength to end the cycle. They read somewhat unsmoothly:

> “Forces displaced continents. Wegener presented evidence from many sciences such that geologists could not . ...”

This manner of retaining a low strength proposition among highly connected propositions was the most common form of incoherence. In such cases, the low strength proposition was not wrong, but irrelevant. Actual changes in meaning were rare. The other major form of incoherence occurred when the central meaning of the gist was lost, such that the overall strength of the propositions was low even after the number of propositions was reduced to the lower memory bound.

Cycle 9. The ninth reduction cycle inputs the final two propositions of the first paragraph. The contents of working memory at the beginning of the cycle are:

P25	DISPLACE [forces, continents]	08
P26	PRESENT [wegener, P27]	78
P27	DERIVE-PROM [evidence, P29]	78
P28	CAUSE [P26, P32]	28
P29	QUANTIFY [sciences, many]	78
P30	POSSESS [wegener, theory]	78
P31	MODAL: could [geologists, P33]	97
P32	NEGATE [P31]	78
P33	MANNER [P34, easy]	97
P34	IGNORE [geologists, P30]	98

As the lowest member of the second reduction category, P25 is deleted first. It was residual to the gist of the propositions even in the previous cycle, and was only left in working memory only as a result of the higher interconnectedness of the other propositions. P28 is then the only remaining member of the second category, and the CORE BEST action simply deletes it from working memory, given that it is not an argument in another proposition. P32 then loses its connections. The reduction rule for NEGATE predicates uses the PRUNE action, deleting all of the nested propositions that are not otherwise connected: in sequence, P32, P31, P33, P34, and P30. The final three propositions, P26, P27, and P29 read:

> “Wegener presented evidence derived from many sciences.”

Note that the PRUNE action deletes the complete set of nested propositions, even though this reduces the number of propositions beneath the limit that would normally end the reduction phase.

Author Notes

Requests for reprints should be sent to Bruce K. Britton, Department of Psychology, University of Georgia, Athens, GA, 30602. The research was supported in part by a grant from the Russell Sage Foundation and Alfred Sloan Foundation Joint Program on Behavioral Economics to the second author, and by a grant from the Air Force Human Resources Laboratory, Learning Abilities Measurement Project (LAMP), Grant AFOSR-89-0515, the Office of Naval Research, Grant 442-804 1---01, and the Office of Educational Research and Improvement, Grant 117A2007 to the third author.

Footnotes

[1] Stating these relationships in equation form, $\underline{S}_{n,0}$ = a, where $\underline{S}$ denotes strength, $\underline{n}$ as the first subscript denotes a new proposition, and the second subscript denotes the number of connections. $\underline{S}_{m,0}$ = (l-a&, where subscript $\underline{m}$ denotes an old proposition, and subscript $\underline{p}$ denotes the final point of the previous reduction cycle. Finally, $\underline{S}_{i+1} = \underline{S}_i + a(1 - \underline{S}_i)$, for both new and old propositions, where subscript $\underline{i}$ indicates the number of connections. If a is set between 0 and 1, then $\underline{S}$ is similarly confined.

[2] Logical relation propositions were set to be protected from reduction when their strength was greater than l-a, so that they would move from the fourth reduction category with protection directly to the second in simulations run using a high a (e.g., .7), from the fourth with protection to the fourth without protection and then to the second category in simulations run with moderate values of a (e.g., .4), and be treated no differently from other coordination predicate propositions in simulations run with low values of a (e.g., .1).

[3] Advances toward simulation of the predicate construction process (e.g., Kieras, 1989; Miller, Beckwith, Fellbaum, Gross, & Miller, 1990) could be joined with the current simulation. It is important to recognize that many predicate classes can only be inferred from a knowledge of the arguments in the proposition. For example, “and” can imply simple conjunction, temporal order, or a causal relationship. Similarly, “of’ can be used to relate arguments in several predicate classes. The inclusion of an automatic propositionalization would give a more precise estimate to the importance the semantic information in predicate categories.

[4] The arguments are then mover and moved, with the REDUCE action applied to the latter, then to the former, followed by the CORE BEST action, favoring the mover.

Table 1

Immediate Resolution Predicates

Category	Predicate Class	Illustration	Reduction Rule
1) References:			
	REFERENCE	"X" refers to Y	CORE non-pronouns if pronoun present. Otherwise CORE NAME prop. Otherwise CORE BEST favor Y.
2) Minimal restriction:			
	HEDGE	approximately X	CORE X.
	INTENSIFY	very X	CORE X.

[illegible]

Fellbaum, Gross, & Miller, 1990) could be joined with the current simulation. It is important to recognize that many predicate classes can only be inferred from a knowledge of the arguments in the proposition. For example, "and" can imply simple conjunction, temporal order, or a causal relationship. Similarly, "of' can be used to relate arguments in several predicate classes. The inclusion of an automatic propositionalization would give a more precise estimate to the importance the semantic information in predicate categories.

[4] The arguments are then mover and moved, with the REDUCE action applied to the latter, then to the former, followed by the CORE BEST action, favoring the mover.

Table 2

Description Predicates

Category	Predicate Class	Illustration	Reduction Rule
3) Modification:			
	QUALIFY	X is Y (adjective)	CORE X.
	MANNER	X was done Y (adverb)	CORE X.
4) Auxiliary verbs:			
	MODAL	can/may/should X	CORE X.
	PROCESS	starts to/continues to X	CORE X.
5) Identification:			
	PART	X is a piece of Y	REDUCE& Otherwise CORE Y.
	NAME	X is named Y	CORE BEST favor Y.
	SET	X is a member of set Y	REDUCE Y. Otherwise CORE X.
	POSSESS	X belongs to Y	REDUCE Y. Otherwise CORE X.
	COMPOSITION	X is made of Y	REDUCE Y. Otherwise CORE&
	IDENTITY	X is a Y	REDUCE Y. Otherwise CORE X.
6) Quantification:			
	QUANTIFY	many/some/few Y	REDUCE EXTENT prop if embedded in it Otherwise REDUCE Y. Otherwise CORE Y.
	NUMBER	X (number of) Y	TRANSFORM to [QUANTIFY: many/some/few Y] and PRUNE&
	RANGE	X_p to X_q (number of) Y	TRANSFORM to [QUANTIFY: many/some/few/?? Y] and PRUNE Xs.
	EXTENT	X (portion or units) of Y	TRANSFORM to [QUANTIFY: many/some/few/?? Y] and PRUNE X. If X is a NUMBER or RANGE prop, REDUCE X first.
7) Negation:			
	NEGATE	not X	If embedded in another prop, then REDUCE that prop. Otherwise PRUNE X.

Note. Predicate classes appear in capitals, reduction rules and conditions in small capital, and arguments in italics.

Table 3

Coordination Predicates

Category	Predicate Class	Illustration	Reduction Rule
8) Communication:			
	COMMUNICATION	X said Y to Z	CORE Y.
	TOPIC	X talked about Y	CORE BEST favor Y.
9) Location in time and space:			
	LOCATION	X at place Y	If embedded in another LOCATION prop, CORE X. Otherwise REDUCE Y. Otherwise REDUCE X. Otherwise CORE X.
	TIME	X at time Y	If embedded in another TIME prop, CORE X. Otherwise REDUCE Y. Otherwise REDUCE& Otherwise CORE X.
10) Numeric/Qualitative/Equality Contrasts:			
	N-CONTRAST	Y holds more/fewer than X	TRANSFORM to [QUANTIFY: many/few Y] and PRUNE&
	Q-CONTRAST	X is more Y than Z is	TRANSFORM to [QUALIFY: X is Y] and PRUNE Z.
	E-CONTRAST	X is like Y	CORE BEST favor&
11) Logical Relations:			
	CAUSE	X led to Y	PROTECT. Otherwise CORE BEST favor Y.
	CONDITION	If X then Y	PROTECT. Otherwise CORE BEST favor Y.
	CONCESSION	X, but Y	PROTECT. Otherwise CORE Y and PRUNE&.
	PURPOSE	X was for Y	PROTECT. Otherwise REDUCE X. Otherwise REDUCE Y. Otherwise CORE Y.
	CONJUNCTION	X and Y (and Z)	PROTECT. Otherwise, drop Z from proposition. Otherwise CORE BEST favor X.

Note. Predicate classes appear in capitals, reduction rules and conditions in small capital, and arguments in italics.

Table 4

Input Accuracy Scores: d’ by Passage, Memory Size, and a

Memory		Passage				
Size	*a*	**Wool**	Drift	Feedback	San Francisco	Average
Large	.1	3.38	3.35	2.82	2.69	3.06
(21, 11, 5)’	.4	3.38	3.35	2.87	2.73	3.08
	.7	3.38	3.35	2.92	2.47	3.04
	Average	3.38	3.35	2.87	2.63	3.06
Moderate-	.1	3.38	3.35	2.51	2.69	2.98
Large	.4	3.38	3.35	2.87	2.83	3.12
(17, 9, 4)[a]	.7	3.38	3.35	2.87	2.73	3.08
	Average	3.38	3.35	2.75	2.75	3.06
Medium	.1	3.14	3.19	2.69	2.95	2.99
(13, 7, 3)’	.4	3.29	3.24	2.77	2.43	2.93
	.7	3.38	3.24	3.14	2.69	3.11
	Average	3.27	3.22	2.87	2.69	3.01
Small	.1	2.63	--	2.91	2.82	2.79
(9, 5, 2)	.4	2.80	2.94	2.98	2.47	2.80
	.7	2.84	2.98	2.31	2.31	2.61
	Average	2.76	2.96	2.73	2.53	2.73

Note. One missing value appears because the model produced no misses.

‘Input memory maximum, reduction-end memory maximum, reductionend memory minimum number of propositions.

Table 5

Memory Prediction: Percent Variance Accounted for bv **Passage,** Memory Size. and a

Memory		Passage				
Size	a	Wool	Drift	Feedback	San Francisco	Average
Large	.1	18.8	18.3	28.2	30.3	23.9
(21, 11, 5)’	.4	21.3	26.5	30.4	23.7	25.5
	.7	29.2	35.7	28.8	23.5	29.1
	Average	23.1	26.8	28.8	25.8	26.1
Moderate-	.1	20.9	22.2	27.8	31.9	25.7
Large	.4	23.1	27.2	35.8	25.3	27.9
(17, 9, 4)’	.7	28.5	35.8	34.1	23.7	30.5
	Average	24.2	28.4	32.6	27.0	28.0
Medium	.1	22.5	20.1	26.5	30.6	25.0
(13, 7, 3)’	.4	20.3	24.0	27.5	26.1	24.6
	.7	25.3	24.5	25.2	20.2	23.8
	Average	22.7	22.9	25.2	25.6	24.4
Small	.1	21.4	18.2	26.1	33.0	24.7
(9, 5, 2)[a]	.4	19.4	17.4	28.8	28.2	23.4
	.7	22.3	24.3	24.9	26.7	24.6
	Average	21.0	20.0	26.6	29.3	24.2

‘Input memory maximum, reduction-end memory maximum, reductionend memory cut-off number of propositions.

Table 6

Residual memory scores (observed - expected no. of subjects) by medicate class with moderate-large working memory.

Category:		No Model'		a = .1		a = .4		a = .7	
Predicate Class	n	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Immediate Resolution:									
REFERENCE	8	-10.8	5.6	-5.9	6.6	-1.7	7.9	-2.4	6.6
HEDGE	2	-10.6	3.7	-11.6	4.8	-12.8	6.4	-7.6	4.4
INTENSIFY	5	-16.9	3.0	-10.8	4.7	-6.4	5.9	-2.4	4.9
Overall	15	-12.8	5.3	-8.3	6.1	4.7	7.7	-3.1	5.8
Modification:									
QUALIFY	48	-7.5	8.3	-0.7	9.7	-2.6	9.6	-3.0	9.3
MANNER	7	-4.2	10.0	0.8	3.8	1.8	4.1	0.3	5.6
Overall	55	-4.7	9.8	-0.5	9.2	-2.0	9.2	-2.6	9.0
Auxiliary verbs:									
MODAL	5	-4.2	10.9	-1.0	6.8	-1.0	8.2	-1.0	8.2
PROCESS	3	4.8	6.1	-5.7	9.7	4.5	8.3	-5.4	8.4
Overall	8	-4.4	8.9	-2.8	7.0	-2.3	7.8	-2.6	8.0
Identification:									
PART	9	-3.6	12.8	-2.5	13.4	-3.5	13.8	-3.5	12.7
NAME	6	4.0	11.1	5.7	9.7	6.0	9.8	4.3	8.4
SET	4	13.3	16.3	5.0	11.6	5.7	16.0	7.7	13.8
POSSESS	24	-1.6	14.0	-3.3	11.8	4.8	11.9	-4.5	11.9
COMPOSITION	5	-4.8	9.2	-5.0	13.7	-5.8	12.6	-9.0	10.7
IDENTITY	6	3.6	15.0	3.2	15.0	2.3	10.8	2.0	14.4
Overall	54	0.1	13.9	-1.0	12.4	-1.9	12.5	-2.2	12.3

(Table continues)

Table 6. (continued)

Quantification:									
QUANTIFY	23	-3.2	13.4	-1.0	10.4	-1.1	10.5	-1.5	11.4
NUMBER	7	11.5	15.8	13.5	10.5	11.6	11.4	9.2	10.0
RANGE	1	13.0	--	17.8	--	13.6	--	12.5	--
EXTENT	8	6.4	11.6	2.4	9.9	2.2	11.4	0.1	7.0
Overall	39	1.8	14.5	2.8	11.6	2.1	11.6	1.1	10.9
Negation:	1	-2.3	--	5.6	--	4.5	--	4.9	--
Communication:									
COMMUNICATION	3	-9.2	4.4	-11.6	2.1	-11.3	4.5	-10.6	5.5
TOPIC	4	6.2	20.1	4.4	17.7	2.8	13.2	0.9	10.6
Overall	7	-0.4	16.6	-2.4	15.2	-3.2	12.3	4.0	10.2
Location in time and space:									
LOCATION	26	-3.1	9.5	4.0	7.3	-2.5	8.0	-3.2	8.2
TIME	18	2.3	10.8	1.0	10.3	0.8	9.7	1.7	2.2
Overall	44	-0.9	10.3	-2.0	8.9	-1.1	8.8	-1.2	9.2
Numeric/Qualitative/Equality Contrasts:									
N-CONTRAST	4	3.0	15.4	-2.5	11.8	1.9	12.2	-0.8	11.2
Q-CONTRAST	7	6.8	14.8	6.8	12.1	4.9	15.6	1.9	13.8
E-CONTRAST	2	3.2	4.9	4.5	0.5	3.4	4.5	2.2	6.2
Overall	13	5.1	13.3	3.6	11.2	3.7	12.7	1.1	11.5
Logical Relations:									
CAUSE	11	0.3	14.4	0.0	13.1	-1.2	12.8	0.9	12.6
CONDITION	1	8.7	--	13.8	--	12.3	--	8.5	--
CONCESSION	1	-7.0	--	-0.9	--	3.0	--	4.8	--
PURPOSE	4	8.2	19.6	7.5	16.0	8.8	16.0	10.8	18.1
CONJUNCTION	13	‘-1.2	11.9	1.6	8.5	2.4	9.5	2.2	9.1
Overall	30	0.8	13.5	2.1	112	2.3	11.5	3.2	11.6
Nonclassified:	111	3.3	13.0	1.0	10.9	1.5	10.3	2.2	10.1

‘These scores are calculated by subtracting the mean score for each story from the observed score of each proposition.