Title
Synergy Between Memory and Model-based Processing: Integration Facilitated by Animation

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 28(28)

ISSN
1069-7977

Authors
Lane, Sean M.
Matthews, Robert C.
Sallas, Bill
et al.

Publication Date
2006

Peer reviewed

Synergy Between Memory and Model-based Processing: Integration Facilitated by Animation

Bill Sallas (sallas@LSU.edu)
Department of Psychology, Louisiana State University
Baton Rouge, LA 70803 USA

Robert C. Mathews (psmath@LSU.edu)
Department of Psychology, Louisiana State University
Baton Rouge, LA 70803 USA

Sean M. Lane (slane@LSU.edu)
Department of Psychology, Louisiana State University
Baton Rouge, LA 70803 USA

Ron Sun (rsun@RPI.edu)
Cognitive Science Department, Rensselaer Polytechnic Institute
Troy, NY 12180 USA

Abstract

Domangue, Mathews, Sun, Roussel, and Guidry (2004) trained Ss to generate valid exemplars from an artificial grammar using either memory-based or model-based processing. Their results showed that learning by memory-based processing resulted in fast but inaccurate performance, while model-based learning resulted in slow but accurate performance. Attempts to integrate both types of training did not result in fast and accurate string generation. Fast and accurate performance was achieved by Sun and Mathews (2005) using a computer animated display to train Ss. The current study used a 2x2x2 factorial design to determine why Ss who view an animated display of a diagram of the grammar during study perform well at test. The results suggest that the diagram informs Ss which letters or chunks of letters can appear in each position, as well as where they cannot appear. Animating the diagram focuses attention on the relevant portion of the complex display and leads to the best performance by creating a synergy between memory and model-based processing.

Introduction

There is considerable evidence that humans are capable of learning through two different types of processes (e.g., Mathews, et al., 1989; Reber, 1969). Explicit learning has been characterized as the conscious and effortful acquisition of rules (Reber, 1993). An example is learning to solve an algebra problem by following a series of steps. Implicit learning has been described as the non-conscious, automatic acquisition of information (Reber, 1969). For example, infants learning to produce novel utterances without an explicit understanding of the rules used to generate those utterances would involve implicit learning (Dienes, Broadbent, & Berry, 1991).

The interpretation of evidence for these two types of processes has been questioned by some researchers. For instance, Shanks & Cannon (2002) have demonstrated that implicit learning tasks can be affected by a secondary task, and argue that this suggests learning in these tasks cannot be considered unconscious. Although this debate continues, Mathews and colleagues (1997; Mathews, et al., 1989) have suggested the focus of the debate has obscured two important issues. First, the term *implicit* has focused too much on the nonconscious aspects of this type of learning. Instead, they suggest implicit learning is similar to pattern recognition. The act of learning involves conscious awareness of a stimulus, although people are often unaware of the features they are encoding which will serve as the basis for later recognition. They propose using the terms *memory-based* (for implicit learning) and *model-based* (for explicit learning) processing. During exposure to exemplars, memory-based processing automatically abstracts patterns of covariance needed to respond appropriately to the task being performed. Model-based processing involves using an explicit representation of the task to guide action (e.g., following a recipe while cooking). A second issue concerns the lack of research regarding how these two processes interact. There is growing acknowledgement that it is difficult, if not impossible to isolate one process in a particular experimental task (e.g., Reber, 1993). Thus, both types of processes interact in both laboratory and real-world settings. For example, when a radiologist views an image looking for cancerous cells, memory-based processing may draw their attention to suspicious looking cells. At the same time, the physician may use model-based processing to consult a list of characteristics which cancerous cells must possess. The issue of interaction, and particularly situations where this interaction is facilitative (*synergistic*), is the focus of the following experiment.

One of the first demonstrations of synergy in the artificial grammar paradigm was provided by Mathews, et al. (1989) who argued that memory-based and model-based processing can interact in positive ways (synergistically). Ss who first viewed exemplars from a bi-conditional artificial grammar (memory-based processing) and then corrected letter strings which contained errors (model-based processing), performed better on a grammaticality judgment test than Ss who received the training in the opposite order or who received only one type of training. Additionally, when a finite-state grammar was used, where the rules are more difficult to generate than the relatively simple logical rules of a bi-conditional grammar, no synergy between memory-based and model-based processing was found. Therefore, when the rules were relatively easy for Ss to generate, exposure to many exemplars (memory-based processing) followed by a task which encouraged model-based processing resulted in a synergy between the two types of processing. When the rules were difficult to generate, as in the finite-state grammar, this synergy did not occur.

Domangue et al. (2004) investigated this interaction by exposing Ss to either exemplars from a finite-state grammar (memory-based processing), the diagram (model-based processing), which is simply a visual representation of the rules used to create the exemplars (see Figure 1), or to the exemplars within the context of the diagram (memory and model-based processing) during training.

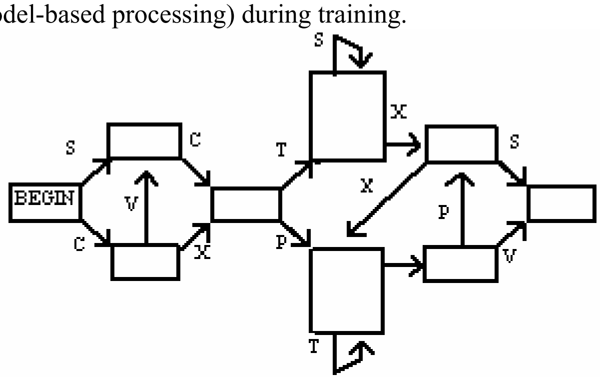

Figure 1. Diagram of grammar used in Domangue et al. (2004).

At test, Ss were required to generate valid exemplars given two cues, or letters. This cued-generation test is different from the grammaticality judgment test which is more commonly used in the artificial grammar literature. Forcing Ss to generate strings makes the task much more like the real world task of language learning (e.g. rarely are we asked to comment on the grammaticality of a sentence, rather we respond to a sentence, or cue, with our own statement). Note, however, that the greater degree of knowledge required to produce responses means that Ss must undergo longer training periods.

Ss in the memory-based processing condition responded quickly, but were inaccurate, while those in the model-based processing condition were slow, but accurate. The expected synergy between memory and model-based processing in the third condition was not found, as Ss followed the memory-based processing pattern of fast but inaccurate responding.

Unlike Domangue et al. (2004), Sun and Mathews (2005) demonstrated that Ss could combine memory and model-based processing to achieve fast and accurate performance on a cued-generate test by using a computer animated training task. Ss performed a string-edit task in which they were shown an exemplar with one or more errors (i.e. letters in incorrect positions) and instructed to identify the incorrect letters. Ss were assisted in this task by an animated diagram of the grammar in which letters appeared one-by-one in their correct position within the diagram.

This is an interesting finding because the diagram-assist group in Sun and Mathews (2005) received the same information as Ss who traced exemplars through the diagram in Domangue et al. (2004). Both groups viewed valid exemplars in the context of a diagram of the grammar. The main difference was that Ss in Domangue et al. manually copied letters from exemplars into the diagram themselves, while those in Sun and Mathews viewed an animation of the letters appearing in the diagram and used that information to complete the string-edit task.

This difference in performance may have been due, in part, to the way Ss viewed the exemplars. The diagramming task in Domangue et al. (2004) encouraged the parsing of whole exemplars into individual letters and placing those letters in the diagram. While this type of training provided knowledge of how exemplars are constructed, it likely failed to focus attention on encoding whole exemplars. In addition, it placed a constant load on working memory, as Ss needed to hold the letter in memory while looking for the correct node of the diagram to place it. Thus, it made Ss less likely to encode whole exemplars.

Based on an analysis of the tasks used in Domangue et al. (2004) and Sun and Mathews (2005), we hypothesized that enhanced performance in the Sun and Mathews' task was due to the interacting effects of the diagram and animation. Specifically, the diagram provided information about the underlying structure of the grammar, while animation highlighted the relevant portion of the model just as Ss were processing the corresponding element of the exemplar. In the process of testing this hypothesis, we sought to eliminate three simpler explanations of Sun and Mathews' findings.

First, it could be that the design of the string edit task forced Ss to predict which letter would appear in the diagram. Predicting the next letter or letters may have led to a generation effect (Slamecka & Graf, 1978) resulting in better memory for letter strings or chunks of letters seen at training compared to Ss in the Domangue et al. (2004) task where they simply copied the exemplars into the diagram.

Second, Ss in Sun and Mathews (2005) reported "chunking" the letter strings into groups of two and three letters. Servan-Schreiber and Anderson (1990) and Perruchet and Pacteau (1990) have found that exposure to exemplars divided into chunks improves Ss ability to make grammaticality judgments. The speeded nature of the animated task may have encouraged Ss to chunk the exemplars as opposed to viewing them as individual letters in Domangue et al.

The third alternative possibility for the fast and accurate performance in Sun and Mathews (2005) is that their training task was animated, while the task in Domangue et al. (2004) was static pen and paper. While it is impossible to compare performance across studies, it does seem possible that animating the diagram in Sun and Mathews may have had a positive effect on Ss' learning by providing information as it was needed to encode the exemplars. Animation may add a temporal element that is lacking in a static display. By displaying exemplars over time, animation may help Ss to encode dependencies between each letter or chunk. While this information is certainly available in the static display, it may become more salient when animated.

The current study tested the hypothesis that an interaction between the mode of presentation (animation) and the content (diagram) in Sun and Mathews (2005) created a synergy between memory and model-based processing by using a 2x2x2 factorial design, with display type (animated and static), content (diagram and chunk), and prediction (immediate or predictive) as factors. As in Sun and Mathews (2005), during training Ss performed a string-edit task in which they identified incorrect letters with various assist cues to help in this task. In the training task, Ss saw the assist cues either with the letters grouped together in chunks (Servan-Schreiber & Anderson, 1990) or in the context of the diagram. Also, the cues were either static or animated. Finally, the cues were either available immediately or became available only after Ss had edited the string (predictive). In addition to the cued-generation test, a grammaticality judgment test was administered.

Method

Ss and Materials

One-hundred and eighty-seven Ss were recruited for this study. All Ss were undergraduate students at Louisiana State University and were given extra credit for their participation. Due to attrition among subjects over the five-day testing period, group size ranged from 19 to 22 Ss.

The finite-state artificial grammar used by Domangue et al. (2004) and Sun and Mathews (2005) was used in the current study. The grammar generates 177 letter strings using the letters, S, C, V, X, T, and P. The letter strings in the grammar range in length from five to eleven letters.

Design

This study used a 2x2x2 factorial design, with content (diagram or chunks), presentation type (static or animated), and prediction (predictive or immediate) as the between-subject factors. A test-only control was also run.

Procedure

Ss were tested in groups up to 8. Ss completed five 1-hour sessions over the course of one week. Sessions 1-4 consisted of a 20-minute study phase and 20-minute string-generation test. In session five, Ss completed a 20-minute string-generation test, followed by a grammaticality judgment test.

Training Phase

Training was conducted through the use of a computer game in which Ss performed a string-edit task (Sun & Mathews, 2005). Ss were shown a letter string at the bottom of the computer screen and instructed to identify the incorrect letters in that string by clicking on those letters with a mouse. Their score was presented in terms of misses (incorrect letters that they did not identify as such) and false alarms (correct letters identified as incorrect). Ss were encouraged to respond quickly but not sacrifice accuracy for speed. A monetary prize was offered to the participant who made the fewest errors to further emphasize accuracy over speed. Each time a letter string was displayed, it contained between one and four errors randomly generated by the computer at the beginning of each trial.

Each training session involved approximately 88 trials. A subset of 22 exemplars from the corpus was randomly selected for each participant by the computer at the beginning of each session. Ss viewed each exemplar four times during the study phase.

Like Sun and Mathews (2005), Ss were given assistance cues to complete the string-edit task. In the static diagram immediate condition Ss saw a diagram of the grammar used to generate the exemplars in the middle of the screen. At the beginning of each trial, all of the letters in the exemplar were shown in their appropriate state in the diagram. Ss could then compare the letters in the diagram to the letters in the string they were editing and mark errors where appropriate. Three seconds after the trial began, a dot appeared under the first letter in the to-be-edited string. After 500ms, the dot moved to the next letter in the to-be-edited string and the participant was no longer allowed to mark the first letter as incorrect. After another 500 ms interval, the dot would move to the third letter in the string and the participant was no longer allowed to edit the second letter, and so on. Thus, the dot was a visual timing device which forced Ss to make quick decisions in the edit task. After the dot passed a letter, an unmarked error was recorded as a miss, and an "X" mark appeared over that letter, alerting Ss of their error. False alarms were also marked with an "X". The timing dot and "X" marks for errors were consistent across all condition. At the end of each trial, the exemplar was displayed at the top of the screen.

In the static diagram predictive condition, Ss performed the string-edit task while the diagram was displayed on the screen. At the end of each trial, all of the letters from the exemplar appeared simultaneously in their appropriate state within the diagram and remained for 3 s.

Ss in the animated diagram immediate condition saw the letters appear one-by-one in the diagram. After a letter appeared in the diagram, Ss had 500 ms to compare that letter to the corresponding letter in the string they were editing and mark an error if necessary. No predictions needed to be made as Ss were able to directly compare letters in the to-be-edited string to the correct letters in the diagram.

Ss in the animated diagram predictive condition also saw the letters appear one-by-one in the diagram. However, they were forced to make predictions about which letter would appear next in the diagram because after a letter appeared in the diagram, Ss were not able to mark the corresponding letter in the to-be-edited string as an error. The letters appeared in the diagram at 500 ms intervals.

The remaining four conditions performed the same string-edit task. In the chunk conditions, the letters from the exemplar appeared in the center of the screen, from left to right, in chunks of two or three letters rather than in a diagram of the grammar. Space was left between the chunks to make them more salient.

Ss in the static chunk immediate condition saw the entire exemplar, segmented into chunks, at the beginning of each trial. Like the static diagram immediate condition, Ss were able to compare letters in the string they were editing to the letters that appeared in the chunked exemplar. Also, like all other conditions, a dot appeared under each letter in the to-be-edited string to alert Ss that their time to mark an error in the letter position was almost over.

Ss in the static chunk predictive condition saw only the to-be-edited string at the beginning of each trial. At the end of the trial the chunked letter string appeared.

In the animated chunk immediate condition, each chunk appeared one at a time, from left to right. As the chunks appeared on the screen, Ss were able to compare letters in the to-be-edited string to those in the exemplar and mark any incorrect letters. After a chunk appeared, Ss had 500 ms per letter in the chunk to mark an error in the corresponding letters in the to-be-edited string.

Finally, in the animated chunk predictive condition, the chunks appeared one at a time. If the first chunk in the exemplar was three letters in length, the timing dot would move through the first three letters in order, stopping for 500 ms at each letter. When the dot moved to the fourth letter, the first chunk would appear. Thus, Ss were forced to predict what letters would appear in the first chunk before seeing the correct chunk of letters appear. Immediate feedback was given if Ss marked a correct letter in the to-be-edited string as correct, or if an incorrect letter was not marked after the timing dot had passed.

Cued-generation Test

At the beginning of each test trial, the computer randomly selected a target exemplar. A set of dashes were displayed on the screen, with one dash for each letter in the target. Two letters, the cues, were displayed in their correct position above the appropriate dashes. Working from left to right, Ss typed letters, one for each dash. The computer then compared the letters that the participant entered with all of the not-yet-generated exemplars in the database. If the string entered by the participant did not match at least 70% of the letters in a valid exemplar, the computer erased any incorrect letters, leaving only letters that matched the closest valid exemplar left in the database. Ss continued this process until at least 70% of the letters matched a not-yet-generated- exemplar. Exemplars have letters in common so it was not necessary for the participant to type the target exemplar chosen by the computer. When the participant reached the 70% criterion, a feedback screen appeared in which the letter string generated by the participant was displayed along with the closest matching exemplar. Once one of the exemplars had been produced, it was removed from the database and could not be generated again until the next test session.

Grammaticality Judgment Test

After the cued-generation test, Ss completed a grammaticality judgment test. At this point, the Ss were instructed that the letter strings that they had seen during the past four sessions followed a set of rules (Reber 1969). They were told that they would see letter strings on the computer screen and they should press one key if the letter string followed those rules and another key if the letter string did not follow the rules.

The grammaticality judgment test consisted of 100 valid exemplars and 140 invalid lures, which could be divided into two groups. One type of lure was created by substituting one intact chunk for another. In some cases, the new chunk came from the same position in the exemplar (i.e. substituting one beginning chunk with another that could not be followed by the rest of the exemplar). In other cases a chunk was replaced with a chunk from a different location (i.e. a beginning chunk replaced by a chunk from the end of an exemplar.) The second type of lure was created by changing one or all of the letters in a chunk to make it invalid.

Results

While ANOVAs were run on all data, only significant results are presented below.

Accuracy

Accuracy was a measure of the proportion of letter strings generated on the first attempt that matched 100% of the letters in the target exemplar per minute in the cued-generate test. A three-way between-subjects ANOVA was run on the accuracy data from the cued-generation test. There was a significant main effect of display content, $F(1, 158) = 4.78$, $p < .05$. Ss who saw the diagram at training produced a greater number of perfect exemplars on the first attempt ($M = .62$) than those who saw chunks at training ($M = .37$).

The ANOVA also revealed a significant interaction between display type and content, $F(1, 158) = 4.23$, $p < .05$. Follow up tests of simple effects revealed that Ss who viewed the animated diagram generated more perfect strings per minute ($M = .84$) than those who viewed the animated chunk display ($M = .34$), $F(1, 77) = 6.5$, $p < .05$. When the display was static, Ss who viewed the diagram (*M* = .41) did not generate significantly more perfect strings than Ss who viewed chunks (*M* = .39), $F(1, 85) = .016$, *ns*.

Speed

Speed was a measure of the total number of attempts made per minute during the 20-minute cued-generation test. A three-way between-subjects analysis of variance (ANOVA) was run on the speed scores. There was a trend towards a main effect of content, $F(1, 158) = 3.75$, $p = .055$, where Ss who saw the diagram at training (M = 8.56) were nominally faster than those who saw the chunks (M= 7.7).

Grammaticality Judgment

Overall grammaticality judgment accuracy was a proportion of the number of letter strings correctly classified as valid divided by the total number of items. A three-way, between-subjects ANOVA was run on accuracy data from the grammaticality judgment test. There was a main effect of content, $F(1, 158) = 4.42$, $p < .05$. Ss who saw chunks in training (*M* = .67) were more accurate than those who saw the diagram at training (*M* = .63).

We next looked at performance on different lure types. When a chunk was in the wrong position, there was a significant main effect of content, $F(1, 158) = 10.89$, $p < .01$. Ss who saw a diagram at training (*M* = .58) were more accurate than Ss who saw the exemplars parsed into chunks (*M* = .52). While there were no significant interactions, there was a trend towards an interaction between content and display $F(1, 158) = 3.32$, $p = .07$. Ss who viewed the animated diagram were nominally more accurate (M = .60) than those who viewed the static diagram (M = .55). There was no difference between Ss who saw the animated chunks (M = .52) and those who saw the static chunks (M = .53).

When there was an error within a chunk, a different pattern of results emerged. There was a significant main effect of content $F(1, 158) = 10.24$, $p = .01$. In this case, Ss who saw the exemplars parsed into chunks at training (*M* = .75) were more accurate than Ss who saw a diagram (*M* = .67). There were no significant interactions.

Discussion

The present experiment replicated the finding that training which combines memory and model-based processing can lead to synergistic effects on learning a finite-state grammar (Sun & Mathews, 2005). More importantly, our results clearly point to the combined role of grammar structure (diagram) and animation as key to obtaining this outcome. Further, our results allowed us to rule out three simpler explanations of our findings.

The first alternative hypothesis was that the animated diagram only provided correct chunk dependency information that was not explicitly available in other conditions. If this hypothesis were true Ss who viewed the exemplars parsed into chunks should have been as accurate on the cued-generation test as those who view the diagram in training. This is because the chunked exemplars shown in training provided the same correct chunk dependency information that was available in the diagram. The cued generate test results showed that viewing the exemplars within the context of the diagram at training led to greater accuracy at test. This main effect of display content was qualified by a content by type interaction where the diagram produced greater accuracy than the chunks only when animated. This means that the utility of the diagram is not just in providing information about correct chunk dependencies. If that were the case, Ss who viewed the chunks should have been as accurate. At least when animated, the diagram provides something more than just information on chunk dependencies.

A second alternative hypothesis was that the animated diagram in Sun and Mathews (2005) encouraged generation (Slamecka & Graf, 1978) because Ss were required to make predictions about which letter would appear next in the diagram, rather than making one-to-one comparisons as in the other conditions. If that were the case, Ss who made predictions in the current study should have been more accurate and faster than those who did not make predictions. The results of the current study show no effect of prediction on speed or accuracy on the cued-generation test.

The third alternative hypothesis was that the temporal element added by animation in Sun and Mathews (2005) may have made the dependencies between letters more salient. If this were the case, Ss in the current study who viewed an animated display should have been more accurate on the cued-generation test than those who viewed a static display. There was no main effect of display type on accuracy during the cued-generation test suggesting that animation by itself did not facilitate learning. Further evidence to reject this hypothesis is that regardless of display type (animated or static), Ss who viewed chunks in training responded accurately on the grammaticality judgment test when errors were of the within–chunk type. Animation across display type did not facilitate learning. Only animation of the diagram facilitated learning.

The overall pattern of results suggests that the utility of the animated diagram does not lie within one factor, as the diagram only produced fast and accurate performance on the cued-generation test when it was animated. The animated diagram provided the same information about correct dependencies between the chunks as the conditions in which the exemplars were parsed into chunks. However, the diagram also provided additional information that was not available in the chunk conditions. Only the diagram, with its pathways between each state, showed Ss what **cannot** come next. Ss in the chunk conditions see that the chunk “TSX” can follow “CVC”, but they are not shown explicitly that “TSX” cannot follow “SCP”. It appears that this information only became salient when the diagram was animated, suggesting that the temporal element provided by animation combined with information about what is and is not allowable was responsible for the accurate performance at test.

The current results suggest that the fast and accurate Ss were not using an explicit model of the grammar during the cued-generation test as in Domangue et al. (2004). The pen-and-paper exemplar diagramming task in Domangue et al. encouraged Ss to parse the exemplars into individual letters and place them in the diagram. Doing so allowed Ss to develop an explicit model of the grammar which resulted in very accurate performance at test. The disadvantage of using the explicit model was slow performance at test.

Unlike Domangue et al. (2004), in the present training task Ss were focused on rapidly perceiving whole (corrected) exemplars. The structured information (diagram or chunks) could be used to correct the target string, but the emphasis of the training task was on producing an intact whole string. Animating the diagram focused Ss' attention on the information relevant at the current point in time. By forcing quick decisions, Ss were encouraged to process the exemplars in chunks rather than in a letter-by-letter fashion. The memory-based processing, developed by processing exemplars in a chunk-by-chunk fashion, combined with model-based processing used to correct the strings, resulted in fast and accurate performance on the cued-generation test. All Ss were fast, because they processed the exemplars in chunks. Ss who saw the static diagram were not as accurate because they could not effectively divide their attention between the edit task and the entire model at the same time. Only Ss who viewed the animated diagram were fast and accurate because they processed the exemplars in chunks and developed knowledge about correct and incorrect placement of the chunks.

The results of the grammaticality judgment test further show that knowledge of chunks (Servan-Schreiber & Anderson, 1990) is not sufficient for accurate performance on this task. When errors were within a chunk, Ss who viewed chunks at training were accurate at identifying those errors. However, when the error was due to a valid chunk placed in the wrong position within the exemplar, those same Ss were no more accurate than the no-training control. We believe this finding suggests that on the more demanding cued generate test, Ss perform best when they have a holistic representation of the valid strings (not just chunks, but including knowledge of how correct chunks are assembled into a whole string).

Previous experiments attempting to integrate memory- and model-based processing to enhance learning in this domain (Domangue, et. al., 2004) failed to obtain the best of both worlds: fast and accurate generation of valid strings. By presenting whole strings while simultaneously running an animated diagram of the grammar this desired synergic effect of employing both types of processing was obtained. We believe the animated diagram allowed Ss to view whole exemplars (needed for memory-based processing) while simultaneously seeing how the string was made (learning about the explicit rules of the grammar) by viewing the relevant part of the diagram as letters appeared in the animation. Apparently, the animated diagram reduced the working memory load that would have been necessary to acquire the same information in a static version of the diagram. We believe this reduced working memory load enabled the two types of learning process to proceed in parallel, resulting in fast and accurate performance on the cued-generate test.

Acknowledgments

This work was supported by an Army Research Institute grant, Contract number W74V8H-04-K-0002, to Ron Sun and Robert Mathews.

References

Craik, F. I. M., & Lockhard, R. S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior, 11,* 671-684.

Dienes, Z., Broadbent, D., Berry, D. (1991). Implicit and explicit knowledge bases in artificial grammar learning. *Journal of Experimental Psychology: Learning, Memory and Cognition, 17,* 875-887.

Domangue, T. J., Mathews, R. C., Sun, R., Roussel, L. G., & Guidry, C. E. (2004). Effects of model-based and memory-based processing on speed and accuracy of grammar string generation. *Journal of Experimental psychology: Learning, Memory, 30,* 1002-1011.

Mathews, R. C., Buss, R. R., Stanley, W. B., Blanchard-Fields, F., Cho, J. R., & Druhan, B. (1989). Role of implicit and explicit processes in learning from examples: A synergistic effect. *Journal of Experimental psychology: Learning, Memory, and Cognition, 15,* 1083-1100.

Perruchet, P., & Pacteau, C. (1990). Synthetic grammar learning: Implicit rule abstraction or explicit fragmentary knowledge? *Journal of Experimental Psychology: General, 119,* 264-275.

Reber, A. S. (1993). *Implict learning and tacit knowledge: An essay on the cognitive unconscious.* New York: Oxford University Press.

Reber, A. S., (1969). Transfer of syntactic structure in synthetic languages. *Journal of Experimental Psychology, 81,* 115-119.

Servan-Schreiber, E., & Anderson, J. R. (1990). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16,* 592-608.

Shanks, D. R., & Cannon, S. (2002). Effects of a secondary task on "implicit" sequence learning: learning or performance? *Psychological Research, 66,* 99-109.

Slamecka, N. J., & Graf, P. (1978). The generation effect: Delineation of a phenomenon. Journal of Experimental Psychology: Human Learning and Memory, 4, 592-604.

Sun, R. and Mathews, R. (2005). *Exploring the Interaction of Implicit and Explicit Processes to Facilitate Individual Skill Learning*. Technical Report TR-1162, Army Research Institute for the Social and Behavioral Sciences, Arlington, VA.