

An Alignment-based Account of Serial Recall

Simon Dennis (Simon.Dennis@colorado.edu)
Institute of Cognitive Science, University of Colorado
Boulder, Co 80301 USA

Abstract

The task of serial recall has become the touchstone for theories of short term memory. Many simple yet powerful computational models have been proposed to account for performance in the task. However, these models typically make only tangential reference to language processes, despite the fact that a key determinant of performance on the task is the extent to which the to-be-learned list mimics the structure of natural language (Miller & Selfridge 1950). The Syntagmatic Paradigmatic model (SP; Dennis submitted, Dennis & Harrington 2001), a memory-based account of sentence processing, is applied to serial recall. Employing an alignment mechanism derived from String Edit Theory (SET; Sankoff & Kruskal 1983), the SP is able to account for the U shaped serial position curve, patterns of interlist and intralist intrusions and the multiply bowed serial position curve that occurs with grouped study lists.

Introduction

The area of short term memory and, in particular, serial recall has engendered a great deal of debate and led to the creation of a number of simple, yet powerful computational models. Current accounts can be divided into chaining models (e.g. Lewandowsky & Murdock 1989) in which list items are assumed to be associated each to the next, ordinal models (Page & Norris 1998) in which list items are assumed to be activated in proportion to their position in the list and positional models (Brown, Preece & Hulme 2000, Henson 1998, Hitch, Burgess, Towse & Culpin 1996) in which list items are each associated with a unique positional cue. Of these three types of models it is the positional models that have been most influential in recent years. In particular, there are now a number of models that propose that temporal cues are generated through oscillatory mechanisms and that it is time rather than position per se that determines performance (Burgess & Hitch 1992; Brown, et. al. 2000).

While these models have been successful over a wide range of data they have been largely silent on the issue of how short term memory interacts with other cognitive processes, most notably with the language system. Since Miller and Selfridge (1950) it has been known that serial lists that mimic the sequential structure of language can induce dramatically increased short term memory spans. This suggests that rather than considering short term memory as a resource employed by the language system, it might be more appropriate to think of serial recall and other short term memory tasks as epiphenomena of the language processing apparatus. If this is the case, then an alignment based approach such as the SP model, which relies on

retrieval from previous linguistic experience, may provide a viable account of serial recall.

In the next section, the SP model is described. Then simulations demonstrating the model's ability to capture the serial position curve, patterns of interlist and intralist intrusions and the serial position curve of grouped lists are presented.

Description of the Syntagmatic Paradigmatic Model

Figure 1 depicts the SP model as it would appear when exposed to the following corpus:

1. John loves Mary
2. Bert loves Ellen
3. Steve loves Jody
4. Who does Bert love ? Ellen
5. Who does Steve love ? Jody
6. When the loud music started John left
7. When the race started Dave left
8. When the lecture started Michael left

The SP model consists of three long-term memory systems, lexical, sequential and relational each of which is defined in terms of syntagmatic and paradigmatic associations. Syntagmatic associations are thought to exist between words that often occur together, as in "run" and "fast". By contrast, paradigmatic associations exist between words that may not appear together but can appear in similar sentential contexts, such as "run" and "walk" (Ervin-Tripp 1970).

Lexical memory consists of a trace for each word comprised of the paradigmatic associates of that word across the corpus. In the example, the lexical trace for John is {Bert, Steve, Dave, Michael} because each of these words substitutes for "John" in a sentential context. For "Bert" and "Steve" the paradigmatic associations derive from the simple active constructions in sentences two and three, while for "Dave" and "Michael" the associations derive from the more complicated clause initial constructions in sentences seven and eight. In the lexical trace, however, these associations are accumulated regardless of their origin.

Sequential memory consists of a trace for each sentence comprised of the syntagmatic associations embodied by that sentence. In the example, the sequential trace for the sentence "John loves Mary" is the string of words, "John", "loves", and "Mary", in order.

Relational memory consists of a trace for each sentence comprised of the paradigmatic associations embodied by that sentence. In the example, the relational trace for "John

loves Mary” would be {John: Bert, Steve; Mary: Ellen, Jody}. Note that although the lexical and relational traces both contain paradigmatic associations, the lexical trace is accumulated over the entire corpus for an individual word (e.g. in this corpus *John* is bound to the distributed pattern containing *Bert, Steve, Dave* and *Michael*), while the relational trace is a binding of the paradigmatic associations of each of the words in a given sentence, so that within the relational trace for “John loves Mary”, *John* is bound only to *Bert* and *Steve* (not *Dave* or *Michael*).

Note that as suggested above the set containing *Mary, Ellen, and Jody* can be thought of as a representation of the “lovee” role and the set containing *John, Bert* and *Steve* as the “lover” role, so the trace is an extraction of the relational information contained in the sentence. That is, the relational trace captures a form of deep structure

In the SP model, sentence processing is characterized as the retrieval of associative constraints from long-term memory followed by the resolution of these constraints in working memory. Creating an interpretation of a sentence/utterance involves the following steps:

Sequential Retrieval: The current sequence of input words is used to probe sequential memory for traces containing similar sequences of words. In the example, traces four and five; “Who does Bert love? Ellen” and “Who does Steve love? Jody”; are the closest matches to the target sentence “Who does John love? #” and are assigned high probabilities:

0.499 who does bert love ? ellen
 0.499 who does steve love ? jody
 0.001 john loves mary
 ...

In this simple example, the retrieved traces contain many of the same words in the same order and consequently are the best retrieval candidates. In general, however, lexical traces are used to establish structural similarity even in the absence of lexical overlap.

Sequential Resolution: The retrieved sequences are then aligned with the target sentence to determine the appropriate set of paradigmatic associates for each word. At this stage, sentential context will affect the trace words that are aligned with each of the input words:

who: who (0.997)
 does: does (0.997)
 john: steve (0.478) bert (0.478)
 love: love (0.998)
 ?: ? (0.998)
 #: jody (0.460) ellen (0.460)

The “#” symbol indicates an empty slot. Ultimately, it will contain the answer to the question. The numbers in brackets

are probabilities associated with the words immediately preceding them. Space precludes a description of how these probabilities are calculated but a full exposition is available in Dennis (submitted). Note that the slot adjacent to the “#” symbol contains the pattern {Jody, Ellen}. This pattern represents the role that the answer to the question must fill (i.e. the answer is the lovee).

Relational Retrieval: The bindings of input words to the corresponding sets of paradigmatic associates (the relational representation of the target sentence) are then used to probe relational long-term memory. In this case, trace one is favoured as it involves the same role filler bindings as the target sentence. That is, it contains a binding of John onto the {Steve, Bert} pattern and it also contains the {Jody, Ellen} pattern.

0.687 john: bert (0.298) steve (0.298)
 mary: ellen (0.307) jody (0.307)
 0.089 bert: steve (0.319) john (0.226)
 ellen: jody (0.320) mary (0.235)
 0.089 steve: bert (0.319) john (0.226)
 jody: ellen (0.320) mary (0.235)

...
 Despite the fact that “John loves Mary” has a different surface form than “Who does John love ? #” it contains similar relational information and consequently has a high retrieval probability.

Relational Resolution: Finally, the paradigmatic associations in the retrieved relational traces are used to update working memory:

who: who (0.997)
 does: does (0.997)
 john: john (0.500) steve (0.488) bert (0.488)
 love: love (0.998) loves (0.153)
 ?: ? (0.998)
 #: mary (0.558) ellen (0.523) jody (0.523)

In the relational trace for “John loves Mary”, “Mary” is bound to the {Ellen, Jody} pattern. Consequently, there is a strong probability that “Mary” should align with the “#” symbol which as a consequence of sequential retrieval is also aligned with the {Ellen, Jody} pattern. Note that the model has now answered the question - it was Mary who was loved by John.

To summarize, the model hypothesizes four basic steps. Firstly, the series of words in the target sentence is used to retrieve traces that are similar from sequential long term memory. Then, the retrieved sequential traces are aligned with the input sentence to create a relational interpretation of the sentence based on the word order. This interpretation is then used to retrieve similar traces from relational long term memory. Finally, working memory is updated to reflect the paradigmatic constraints retrieved in the previous step.

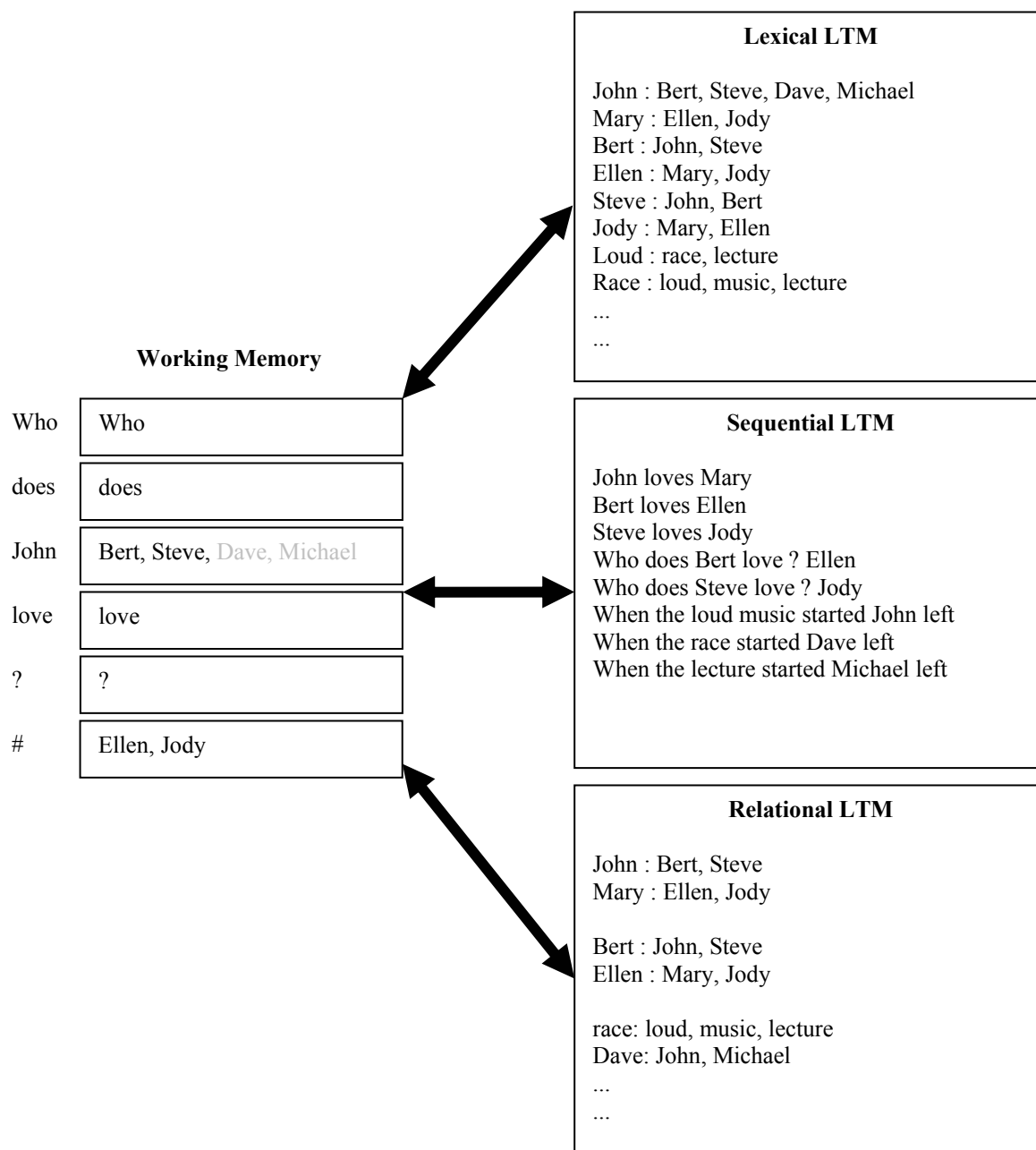


Figure 1: The SP Model architecture. At the start of processing the lexical items are used to retrieve from lexical memory. At this stage, the John slot is filled with Bert, Steve, Dave and Michael. However, only Bert and Steve are context relevant (based on this corpus), and so after sequential retrieval Dave and Michael are eliminated.

In a number of circumstances, it is necessary for the model to be able to distinguish between traces that were stored in the current context from those that are part of the background memory of the system. Rather than propose a separate memory system to store the recent traces, the SP model assumes that these traces are more available because they contain a representation of the current context. During retrieval they are favoured while the same context is in

effect. The content and control of context is poorly understood (Dennis & Humphreys 2001). Rather than try to provide explicit context processing mechanisms, the model simply uses a symbol (CC, C1, C2, ...) to represent the appropriate context and otherwise treats these symbols as if they were words. When a given retrieval probe shares context with traces in memory the same context symbol is used in each. In this paper, the contextual mechanism will

be used to isolate the previous study list as the one to be recalled. This treatment of context is somewhat arbitrary, but is used here in lieu of a more comprehensive mechanism.

The above description provides an overview of the model. However, important issues such as how one compares word sequences of different lengths and decide upon appropriate alignments have not been addressed. Fortunately, there exists a well established literature called String Edit Theory (SET) which provides mathematical foundations for these decisions. Dennis (submitted) provides an exposition of the Bayesian version of SET employed by the SP model as well as mechanisms for comparison and alignment of relational traces. In the next section, we describe the dataset used to test the model.

The Serial Position Curve

In creating an SP account of any given phenomena we must first determine what previous experience is likely to be driving performance. In the serial recall task, it is presumably our experience with lists such as phone numbers, shopping lists etc that provide the traces upon which the control of the task depends. Suppose for instance that the sequential memory system contains the following traces:

1. C1 study the following list , bread milk shampoo fruit meat toothpaste .
2. C2 study the following list , Bill Mary Peter Harry Sue Bert .
3. C3 study the following list , oak gum willow birch pine aspen .
4. C1 recall the items bread milk shampoo fruit meat toothpaste .
5. C2 recall the items Bill Mary Peter Harry Sue Bert .
6. C3 recall the items oak gum willow birch pine aspen .
7. C4 study the following list , 1 2 3 4 5 6 7 .

C1-C4 represent context markers designed to isolate the list that must be recalled. As the current context is always used as a retrieval cue, traces from the corresponding study list are more available at recall (i.e. are more likely to be retrieved) than other traces.

Each of the traces is either an instance of studying a list or recalling a list. For the purposes of the example, quite stylized instructions have been used (i.e. “study the following list” or “recall the items”). It is assumed that a much broader set of possible utterances would be available and that the lexical system would allow the model to identify alternative ways of invoking the same process.

Note that there is no recall instance in the C4 context. This is the list that the model will be required to recall and so we probe the model with the following string “C4 recall the items #####.” The items in the list have been labelled 1 through 7. Note, however, that this labelling is purely to facilitate interpretation of the results. The model only has access to the traces listed above and therefore has no

background knowledge that would allow it to identify the numerical ordering of these labels.

Figure 2 shows the working memory representation following relational retrieval in the serial recall task and Figure 3 shows the probability of retrieval (i.e. the substitution probability of the correct item) as a function of serial position following relational retrieval.

```

c4: c4 (.99) c1 (.30) c2 (.30) c3 (.30)
recall: recall (1.00)
the: the (1.00)
items: items (1.00)
#: 1 (.59) bread (.22) bill (.22) oak (.22) 2 (.16) 3 (.05) milk (.03) mary (.03) gum (.03) list (.02) 4 (.02) , (.02) following (.02) the (.01) . (.01) study (.01) c4 (.01) 5 (.01) 7 (.01)
#: 2 (.38) 3 (.19) milk (.17) mary (.17) gum (.17) 1 (.17) 4 (.07) shampoo (.05) peter (.05) willow (.05) bread (.03) bill (.03) oak (.03) 5 (.03) , (.02) list (.02) following (.02) the (.02) . (.02)
#: 3 (.31) 4 (.19) 2 (.19) shampoo (.14) peter (.14) willow (.14) 5 (.08) 1 (.05) cheese (.05) sue (.05) pine (.05) milk (.05) mary (.05) gum (.05) 6 (.03) , (.02) list (.02) following (.02) the (.02)
#: 4 (.29) 3 (.19) 5 (.19) cheese (.14) sue (.14) pine (.14) 2 (.07) 6 (.07) shampoo (.05) peter (.05) willow (.05) tomatoes (.05) tom (.05) fir (.05) 1 (.02) 7 (.02) , (.02) list (.02) the (.02)
#: 5 (.31) 4 (.19) 6 (.19) tomatoes (.14) tom (.14) fir (.14) 3 (.08) cheese (.05) sue (.05) pine (.05) 7 (.05) meat (.05) jack (.05) beech (.05) 2 (.03) . (.02) , (.02) the (.02) list (.02)
#: 6 (.38) 5 (.19) meat (.17) jack (.17) beech (.17) 7 (.17) 4 (.07) tomatoes (.05) tom (.05) fir (.05) carrots (.03) ruby (.03) aspen (.03) 3 (.03) . (.02) the (.02) list (.02) following (.02) study (.02)
#: 7 (.60) carrots (.22) ruby (.22) aspen (.22) 6 (.16) 5 (.05) meat (.03) jack (.03) beech (.03) 4 (.02) . (.02) the (.01) following (.01) list (.01) study (.01) , (.01) c4 (.01) 3 (.01) 1 (.01)
.: . (1.00)

```

Figure 2: Working memory following relational retrieval in the serial recall task.

The serial position graph shows both the primary and recency components which are the hallmark of the serial recall task, although it tends to overestimate the size of the recency effect when compared to human data (Hitch, Burgess, Towse & Culpin 1996). To understand why the SP model produces the serial position curve consider the sequential representation formed during processing of the study list (Figure 4).

Note that the vectors associated with the start and end of the list have stronger representations than those closer to the middle of the list and therefore are stronger cues at test. The reason is that the instruction words and the end of the sentence form anchors in the study list because they match in each of the retrieved sequential traces. Because alignments containing long contiguous gaps are preferred over alignments containing many short gaps, positional uncertainty increases as you move away from the anchor points. This compromises performance in the middle locations generating the serial position curve. The exaggerated recency effect shown by the model may occur because in the simulation the end of list marker (i.e. the

period) is provided as part of the cue. In reality, however, subjects would have to project forward the location of the end-of-list marker making its location uncertain. This uncertainty would tend to decrease performance at final positions.

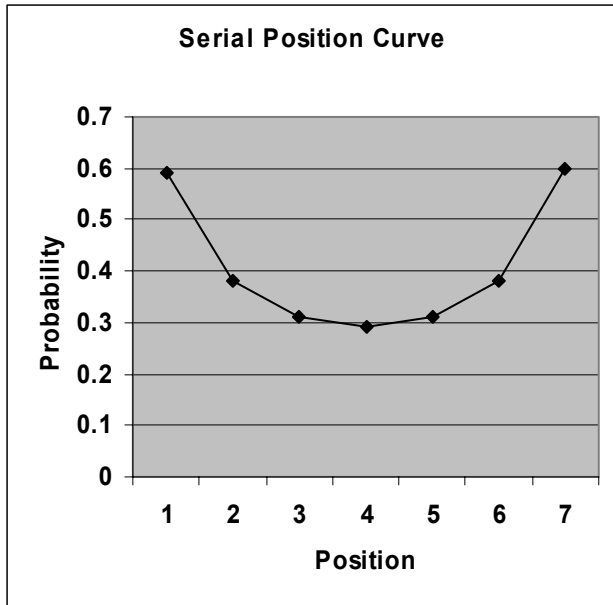


Figure 3: The serial position curve of the SP model.

c4: c1 (.30) c2 (.30) c3 (.30)
 study: study (1.00)
 the: the (1.00)
 following: following (1.00)
 list: list (1.00)
 ∴ (1.00)
 1: bread (.22) bill (.22) oak (.22) milk (.03) mary (.03) gum (.03)
 2: milk (.17) mary (.17) gum (.17) shampoo (.05) peter (.05) willow (.05)
 bread (.03) bill (.03) oak (.03)
 3: shampoo (.14) peter (.14) willow (.14) cheese (.05) sue (.05) pine (.05)
 milk (.05) mary (.05) gum (.05)
 4: cheese (.14) sue (.14) pine (.14) shampoo (.05) peter (.05) willow (.05)
 tomatoes (.05) tom (.05) fir (.05)
 5: tomatoes (.14) tom (.14) fir (.14) cheese (.05) sue (.05) pine (.05)
 meat (.05) jack (.05) beech (.05)
 6: meat (.17) jack (.17) beech (.17) tomatoes (.05) tom (.05) fir (.05)
 carrots (.03) ruby (.03) aspen (.03)
 7: carrots (.22) ruby (.22) aspen (.22) meat (.03) jack (.03) beech (.03)
 ∴ (1.00)

Figure 4: The working memory representation at study.

Intra-list Intrusions

In addition to producing the serial position curve, the uncertainty that accumulates as one moves away from the start and end anchors also means that intra list intrusions (i.e. producing an item that did appear on the list in an incorrect position) are more likely in adjacent positions and in the middle of the list (see Figure 5) as is the case in human data.

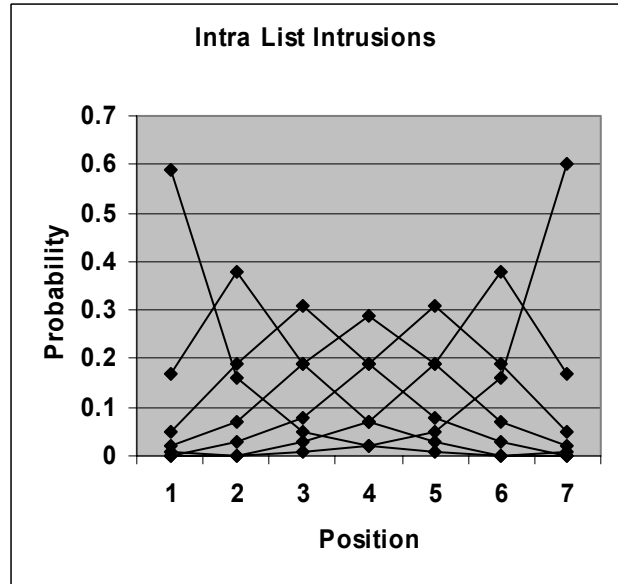


Figure 5: Intra list intrusions in the Serial Recall task.

Inter-list Intrusions

The most relevant traces in memory for the study episode are typically the immediately preceding lists. They are often of equivalent length, are constructed of similar materials and have identical sets of instructions. As a consequence, they are likely to be retrieved during study list processing and will become part of the relational representation of the current list. At test then, there will be a relatively high probability that items from the previous lists will intrude. Figure 6 shows the probability of substitution of items from a given position in each of the previous lists as a function of output position. Note that items are most likely to be output in the same or a similar position to that in which they appeared as is true of human subjects.

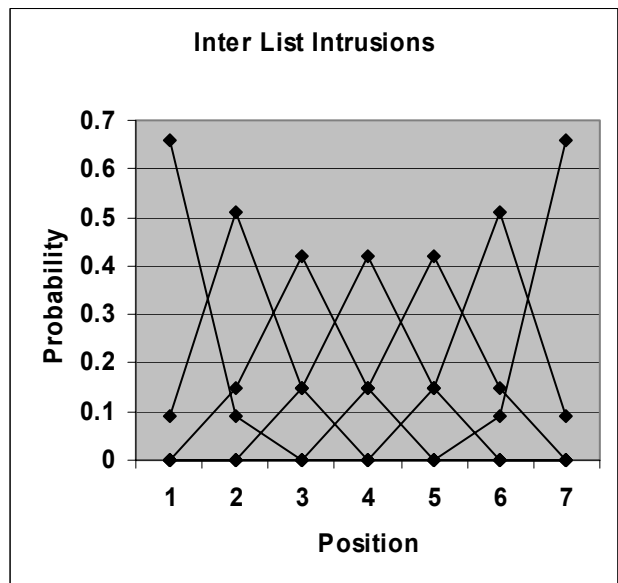


Figure 6: Inter list intrusions in the serial recall task

Serial Position with Grouped Lists

Finally, we consider how the SP model would deal with lists that are grouped using short pauses between groups. Grouped lists have been of recent interest as they allow the effects of timing versus position to be separated (c.f. Ng & Mayberry 2002). In one such experiment (Hitch et. al. 1996) subjects were presented with lists of nine items, each of which was divided into three groups of three with short pauses. As with other similar designs an overall U shaped serial position curve is observed as well as mini-serial position curves within each group. To simulate this data the SP model was exposed to the following corpus:

1. C1 study , oak gum pine .
2. C1 recall . oak gum pine .
3. C2 study , bread milk muffins . shampoo tomatoes beans . meat carrots jam .
4. C2 recall . bread milk muffins . shampoo tomatoes beans . meat carrots jam .
5. C3 study , 1 2 3 . 4 5 6 . 7 8 9 .

The periods embedded in the list represent the pauses. When cued with “C3 recall . # # # . # # # . # # # .” the serial position curve seen in Figure 7 resulted. Again the model reproduces the shape of the human data reasonably well without any parameter fitting, except that the model displays a more pronounced recency effect than is typically the case.

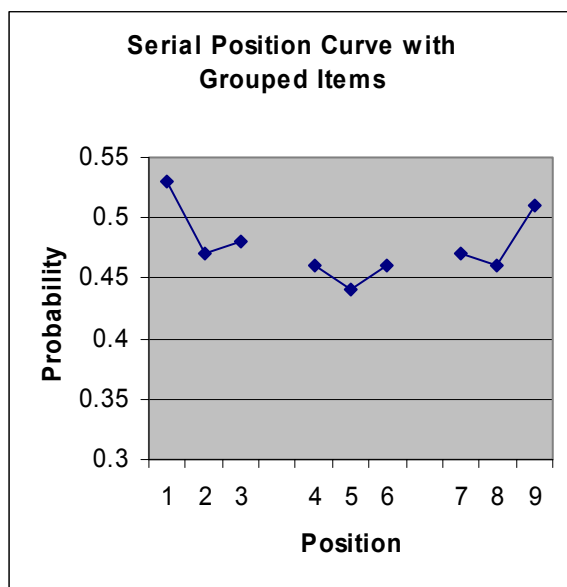


Figure 7: Serial position curve with grouped items

Conclusions

In qualitative terms, then, the SP model seems capable of addressing some key phenomena from the serial recall literature. There are clearly many more phenomena in this domain to which the model must be applied before it can be considered a serious contender as a model of short term memory. In one sense, however, the SP account of short term memory can be seen as an implementation of existing models such as the Start End model (Henson 1998).

In another sense, though, the SP model departs from the normal way of viewing short term memory. Rather than assume that short term memory is a limited resource which the language system employs to carry out sentence processing, the SP model conceptualizes short term memory as an epiphenomena of the language system. If measures of short term memory are reliable indicators of linguistic performance it is because they are measuring linguistic capability directly, rather than because they are quantifying the resources available for linguistic processing. One need not resort to chunking hypotheses to reconcile how different amounts of information of different kinds are retained in a fixed capacity store, but can appeal to the varying levels of support these structures receive from sequential long term memory to account for span variability. While much remains to be done to substantiate such a view, the SP model offers a blueprint for how the areas of short term memory and sentence processing might be more fundamentally integrated than is currently the case.

Acknowledgments

This research was supported by Australian Research Council grant A00106012, US National Science Foundation grant EIA-0121201 and US Department of Education grant R305G020027.

References

- Brown, G. D. A., Preece, T., & Hulme, C. (2000). Oscillator-based memory for serial order. *Psychological Review*, 107, 127-181.
- Burgess, N. & Hitch, G. J. (1999). Memory for serial order: A network model of the phonological loop & its timing. *Psychological Review*, 106, 551-581.
- Dennis, S. (submitted). A memory-based theory of verbal cognition.
- Ervin-Tripp, S. M. (1970). Substitution, context, and association. In L. Postman and G. Keppel (Ed.), *Norms of word association*. New York: Academic Press, pp. 383-467.
- Henson, R. N. A. (1998). Short-term memory for serial order: The Start End Model. *Cognitive Psychology*, 36, 73-137.
- Hitch, G. J., Burgess, N., Towse, J. N., & Culpin, V. (1996). Temporal grouping effects in immediate recall: A working memory analysis. *Quarterly Journal of Experimental Psychology*, 49A, 116-139.
- Lewandowsky, S., & Murdock, B. B. Jr. (1989). Memory for serial order. *Psychological Review*, 96, 25-57.
- Miller, G. A., & Selfridge, J. A. (1950). Verbal context and the recall of meaningful material. *American Journal of Psychology*, 63, 176-185.
- Page, M. P. A., & Norris, D. G. (1998) The primacy model: A new model of immediate serial recall. *Psychological Review*, 105, 761-781.
- Sankoff, D. & Kruskal, J. B., eds (1983). *Time warps, string edits and macromolecules: the theory and practice of sequence comparison*. Addison Wesley.