The Combinatorial Power of Experience

Brendan T. Johns,[1] Randall K. Jamieson,[2] Matthew J. C. Crump,[3] Michael N. Jones,[4] & Douglas J. K. Mewhort[5]
[1] btjohns@buffalo.edu, Department of Communicative Disorders and Science, University at Buffalo, Buffalo, NY
[2]Department of Psychology, University of Manitoba, Winnipeg, MB
[3]Department of Psychology, Brooklyn College, Brooklyn, NY
[4]Department of Psychological and Brain Sciences, Indiana University, IN
[5]Department of Psychology, Queen's University, Kingston, ON

Abstract

Recent research in the artificial grammar literature has found that a simple explicit model of memory can account for a wide variety of artificial grammar results (Jamieson & Mewhort, 2009, 2010, 2011). This classic of model has been extended to account for natural language sentence processing effects (Johns & Jones, 2015). The current article extends this work to account for sentence production, and demonstrates that the structure of language itself provides sufficient power to generate syntactically correct sentences, even with no higher-level information about language provided to the model.

Keywords: Language production; Computational models of language; Corpus-based models.

Introduction

Human languages are both productive and regular. They are productive in that an infinite number of utterances are possible for any language, and regular in that the utterances produced by speakers are systematic. In order to explain these aspects of language, it has been proposed that it is necessary to have a formal grammar underlying language performance (Chomsky, 1959). A grammar of sufficient complexity can construct utterances of any length while maintaining consistency in utterance construction.

The need and evidence for formal grammars in explaining language performance has been argued in many different places (e.g. Christiansen & Chater, 2008; Evans & Levinson, 2009). The goal of this article is not a rehashing of this debate but instead to examine the interaction between regularity and productivity, cornerstones of any theory of language, with the ultimate goal being to understand the constraints that the structure of language itself provides. Specifically, given that languages are highly regular and consistent, the question that will be examined here is what power this regularity provides to the production of grammatically correct utterances.

Historically, to make the problem manageable, researchers have studied how people learn the grammar of small artificial languages. Classic accounts of artificial grammar results propose that as subjects are exposed to strings randomly generated from a pre-defined grammar, they are capable of using this experience to internally generate a representation of this grammar, and in turn use this abstraction to discriminate between grammatical and ungrammatical strings (Reber, 1967). However, people are unable to verbally describe the rules they used to accomplish this discrimination (i.e., it is implicit knowledge). Recent results by Jamieson and Mewhort (2009, 2010) call this explanation into question. Jamieson and Mewhort demonstrated, across a wide variety of tasks and manipulations that a simple explicit memory model (Minerva 2; Hintzman, 1986) can account for most of the relevant findings in the artificial grammar literature.

The reason why a simple recording of the environment could explain artificial language results is clear: the stimuli contained in an artificial grammar task provide a lot of information about the structure of the underlying grammar. That is, like natural language, the artificial languages that are constructed from artificial grammars are *regular*. This regularity causes the underlying structure of a grammar to emerge across the exemplars that are displayed in an artificial grammar experiment, even with no higher-level abstraction taking place during learning.

To formalize this, Jamieson & Mewhort (2011) have implemented the latter position of a string using a model for memory that combines the storage and retrieval operations from Hintzman's (1986) MINERVA 2 model of episodic memory with the holographic reduced representations from Jones and Mewhort's (2007) BEAGLE model of word meaning. According to the model, each studied grammatical sentence is stored to memory. When a probe is presented at test, it retrieves all of the stored sentences in parallel. If the information retrieved from memory is consistent with the probe, the probe is judged to be grammatical; else, it is judged to be ungrammatical.

Despite the model's simplicity, it predicts a surprising number of results in the artificial-grammar task including (a) the linear relationship between mean judgement accuracy and the redundancy of the generative grammar, (b) judgements of grammaticality for individual test items, (c) grammatical string completion, and (d) variation in peoples' judgements depending on how they represent strings in memory (Chubala & Jamieson, 2013; Jamieson & Mewhort, 2009, 2010, 2011). But, why?

The power of this model comes from the natural correlation between the form and amount of structure in an exemplar produced with a grammar and the grammar that was used to produce it (Jamieson & Mewhort, 2005). It follows, then, that each studied grammatical exemplar provides information about the underlying grammar. It also follows that a collection of grammatical exemplars will almost always provide a sum of information greater than that provided by one exemplar alone. The question, then, is not how information about the grammar can be stored in memory, but how is that information harnessed at retrieval?

In artificial grammar experiments, parallel retrieval of grammatical exemplars is sufficient (Jamieson & Mewhort, 2011). But, can that class of explanation scale from handling small artificial grammars to handling how people learn grammar in their natural language?

The goal of this article is to examine the power that the productive and regular nature of language provides in allowing an exemplar model of natural language to produce grammatically correct utterances. Exemplar models are attractive models to examine this question because the complexity of the models lies in the complexity of the information that is being encoded. By storing natural language sentences within an exemplar memory, and determining how effective this stored information is at generating grammatically correct utterances, an examination of the power of the structure of language is provided.

The Exemplar Production Model (EPM)

Our model combines the storage and retrieval scheme from Hintzman's (1986) MINERVA 2 model of episodic memory with the reduced holographic representation scheme from Jones and Mewhort's (2007) BEAGLE model of semantics, and is similar to an exemplar approach to language comprehension recently explored by Johns & Jones (2015).

Representation

In the model, each word is represented by its own unique random vector, **w**, of dimensionality N, where each dimension takes a randomly sampled value from a normal distribution with mean zero and standard deviation $1/\sqrt{N}$. In the simulations that follow, $N = 1024$.

Any given sentence is represented by two vectors, both of which are constructed from the word representations. The first vector, **c**, is called the sentence's context vector and is computed as,

$$\mathbf{c} = \sum_{i=1}^{n} \mathbf{w}_{\boldsymbol{i}}$$

where **c** is the context vector, n is the number of words in the sentence, and $\mathbf{w_i}$ is the vector that represents the word in serial position i of the sentence. As shown, the context vector sums the information from all of the words that appear in the sentence, but it does not include any information about the order in which the words occurred. For example, the context vector that encodes "eat your dinner" is equal to the context vector that encodes "dinner your eat".

The second vector, **o**, is called the sentence's order vector and is computed as,

$$\mathbf{o} = \sum_{i=1}^{n} \mathbf{w}_i \circledast \mathbf{l}_i + \sum_{i=2}^{n} \mathbf{w}_{i-1} \circledast \mathbf{w}_i + \sum_{i=3}^{n} \mathbf{w}_{i-2} \circledast \mathbf{w}_{i-1} \circledast \mathbf{w}_i$$

where **o** is the order vector, n is the number of words in the sentence, w_i is the word in serial position i, w_{i-1} is the word in serial position $i - 1$, w_{i-2} is the word in serial position $i - 2$, l_i is a vector that represents serial position i, and $\circledast$ denotes directional circular convolution (see Jones & Mewhort, 2007; Plate, 1995). As shown, the order vector sums information about (a) what word appears in each serial position in the sentence (i.e., serial position information), (b) which pairs of words follow one another from left to right in the sentence (i.e., bigram information), and which triplets of words follow one another from left to right in the sentence (i.e., trigram information). Given the inclusion of trigram information, the formula cannot be applied to a sentence with fewer than three words.

Finally, a sentence's vector representation, **s**, is a $2N$ dimensional vector formed by concatenating the N dimensional context vector and the N dimensional order vector such that dimensions $1 \ldots N$ in **s** store the context vector and dimensions $N+1 \ldots 2N$ in **s** store the sentence's order vector. Thus a sentence is represented as a vector **s** that is equal to **c** // **o**, where // represents concatenation.

Storage of language experience

To represent experience with language, we store m sentences to a $m \times 2N$ matrix, where rows represent sentences and columns represent features that encode the information in the sentence. Thus, memory for 1000 sentences is represented in a 1000 × 2048 matrix whereas memory for 125,000 sentences is represented by a 125,000 × 2048 matrix.

Retrieval

Retrieval in the model is probe-specific, similarity- driven, and parallel. When a probe is presented to memory, it interacts with the information in the stored traces to construct the memory of a previously experienced event. Decision follows from the construction. Because retrieval is similarity-driven, a probe retrieves traces that are similar to it. Because a probe retrieves whole traces from memory and these whole traces record both context and order information in a sentence, a probe that includes just the context information will also retrieve the order information that it has co-occurred with in the past. This is how the model simulates cued-recall, and it is the mechanism that the model uses to retrieve a sentence (i.e., word order) given a context vector (i.e., an unordered list of words).

The echo, **e**, is computed as,

$$\mathbf{e} = \sum_{i=1}^{m} \left(\frac{\sum_{j=1}^{N} p_j \times M_{ij}}{\sqrt{\sum_{j=1}^{N} p_j^2}\sqrt{\sum_{j=1}^{N} M_{ij}^2}} \right)^9 \times M_i$$

where **p** is the context vector that encodes an unordered list of words (i.e., includes information in serial positions $1 \ldots N$ with serial positions $N+1 \ldots 2N$ set to zero), **M** is the memory matrix that stores the model's sentence knowledge, **e** is the echo, and m is the number of sentences stored in memory. As with a sentence representation, features $1 \ldots N$ in **e** represent the context vector retrieved from memory and features $N+1 \ldots 2N$ in **e** represents the order vector retrieved from memory. Although it is not explicitly stated in the formula, if the similarity between a probe and memory trace is less than zero, the similarity is rewritten as equal to 0 (i.e., equivalent to excluding the trace from the calculation of the echo).

Decision

Our goal is to measure the model's ability to produce a syntactic sentence composed of words presented in an unordered words list. For example, given the words eat, dinner, and your, we would like the model to produce "eat your dinner" rather than "dinner eat your".

To accomplish the transformation from unordered word list to syntactic production, the model compares the order vector in the echo to each of the *n!* order vectors corresponding to the *n!* ways or ordering the words in the unordered list. For example, given the list eat, your, and dinner the model retrieves an order vector based on the context vector, $\mathbf{c} = \mathbf{w}_{eat} + \mathbf{w}_{your} + \mathbf{w}_{dinner}$, and then compares the retrieved order vector against all 3! = 6 sentences that can be constructed from the three words: "eat your dinner", "eat dinner your", "your eat dinner", "your dinner eat", "dinner eat your", and "dinner your eat". The order vector that is most similar to the information in the echo is selected as the best alternative. Because all other orders bear some similarity to the order information in the echo, the operation can also be used to rank order the model's preference over all possible *n!* sentences from first (i.e., most similar) to last (i.e., least similar).

Summary

In summary, the model builds and stores a representation of each sentence in a text corpus. When presented with an unordered word list, the model retrieves a corresponding order vector and produces a word order that corresponds to the order vector that is most similar to the order vector retrieved from memory.

Simulations

The simulations that follow apply the model to a sentence production task. Each simulation involved two major steps. First, we constructed a record of language experience by storing *m* sentences of length *n* to memory; the sentences were sampled randomly from a corpus. Second, we computed the model's ability to translate each of 200 unordered word lists of length *n* into ordered sentences of length *n*.

We expect the model will re-write unordered word lists as syntactic sentences. If true, our simulations would demonstrate that parallel retrieval from a record of language is sufficient to produce at least one hallmark of syntactic behavior. This would reinforce the power of exemplar models of language, and would add to the growing literature on the importance of individual experience with language (Abbot-Smith & Tomasello, 2006; Jamieson & Mewhort, 2010, 2011; Johns & Jones, 2015).

Sentences

Given our goal is to conduct an analysis of natural language, it was critical that we get a fair sample of natural language use. To meet that demand and to model a wide range of language experience, we assembled a pool of 6,000,000 sentences from a number of sources including Wikipedia articles, Amazon product descriptions (McAuley & Leskovec, 2013), 1000 fiction books, 1050 non-fiction books, and 1500 young adult books. Once collated, we edited the list to exclude repetitions and, then, we organized the total list into sub-lists of sentences composed of 3, 4, 5, 6, and 7 words. Finally, we used the sentences in the final pool to construct a list of 200 three word test sentences, 200 four word test sentences, 200 five word test sentences, 200 six word test sentences, and 200 seven word test sentences. All sentences simple in construction, and use mostly high frequency word, but given the complexity of the task provide a useful assessment of the model's performance (all sentences used in the below simulations can be found at http://btjohns.com/experience_sents.txt). No general syntactic construction was used, but the majority consist of single phrase structures. Examples of sentences used for each sentence size are:

3 - *Eat your dinner*.
4 - *The children were flourishing*.
5 - *He walked into the bedroom*.
6 - *She held something in one hand*.

Simulation parameters

We conducted simulations as a function of two key parameters: sentence length (i.e., *n*) and language experience (i.e., *m*). Analysis of sentence length was accomplished by conducting separate simulations for sentences of length *n* = 3, 4, 5, 6, and 7. Analysis of language experience was accomplished by conducting separate simulations given *m* = 1000, 2000, 3000 … 125,000 sentences stored in memory. To ensure that results were not conditional on a particular record of language experience, each simulation was conducted using a different random sample of *m* words. Crossing both factors produces a 5 (sentence length) x 125 (language experience) factorial design.

Two measurements of performance

We measured sentence completion performance two ways. The first method tallied the percentage of tests in which the model most preferred the word order that corresponded to the original sentence.

The second method ranked the model's decisions for all possible word orders from first (i.e., most similar to the order vector in the echo) to last (i.e., least similar to the order vector in the echo) and, then, recording the rank at which the original input sentence appeared. For example, if the model was given "eat your dinner" it would produce a rank order of all six possible sentences composed of the three input words. If "eat your dinner" was the third preferred word order, the trial would be scored as a rank 3 decision.

In summary, the first percent correct measure was an absolute index of model performance, as if the model (like an experimental subject) provided a single response for each test sentence. The second rank based measure offers a more nuanced assessment that measures how close the model was to making the right decision whether its first choice matched or did not match the exact word order in the test sentence.

Results

Figure 1 shows results over the complete 5 × 125 factorial design. The top panel in Figure 1 shows the percent correct production rate (e.g., the model returned "eat your dinner" when presented with "eat your dinner"). The bottom panel in Figure 1 shows the mean rank of the model's decisions. In both cases, performance for 3, 4, 5, 6, and 7 words test sentences are presented as different lines with language experience measured in number of sentences ordered along the abscissa.

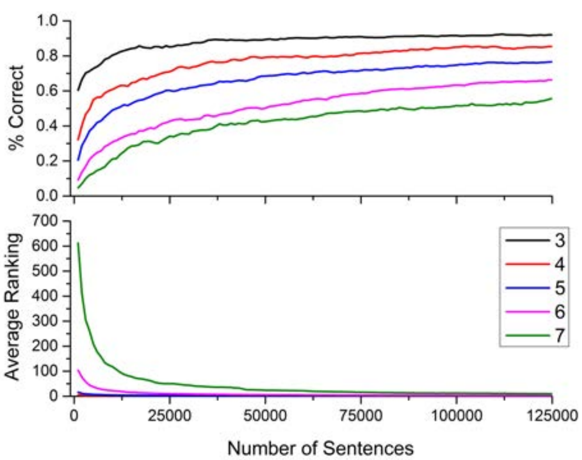

Figure 1. Syntactic completion rates. Top panel shows the percentage of items that the model reproduced the exact word order in the test sentence. Bottom panel shows the mean rank order of the model's preference for the exact word order in the test sentence where 1 = best.

As shown in the top panel of Figure 1, the model's ability to reproduce word order in the presented test sentence varies as a monotone function of sentence length, being best for short sentences and worse for long sentences. However, one must exercise caution in making that comparison. Chance performance changes dramatically as a function of sentence length. At the extremes, when n = 3 the probability of guessing the correct word order is equal to 1 in 6 whereas chance is equal to 1 in 5040 when n = 7. Thus, performance at m = 1000, the final score of 90% correct in the three word condition might be considered as or even less impressive than the corresponding but lower score of 52% correct in the seven word condition.

Also shown, the model's performance improves as a function of language experience, with the improvement being fastest early on in the addition of sentences to memory and slowing considerably after memory includes a record of approximately 50,000 sentences.

The bottom panel in Figure 1 presents a more nuanced picture of results. As shown, when language experience is modest, m = 1000, the model commits far misses (i.e., large mean rank scores). But as language experience increases, mean rank scores drop considerably to nearly 1 for most sentence lengths, and under 10 for all. Considering the results in the top and bottom figures shows that even though the model does not always choose the particular word order in the input sentence, it nevertheless has a strong preference for that specific word order. So, what does a near miss mean?

Although we have scored near misses as wrong, they may occasionally be correct in the broader sense. For example, consider that the model preferred "they quietly went down the stairs" when tested on "they went quietly down the stairs". Although the model failed to produce the input sentence, it nevertheless generated a syntactic alternative.

Figure 2 shows results when m = 125,000 sentences (hatched bars) and when m = 500,000 sentences (closed bars). The top panel shows results with the percent correct measure; the bottom panel shows results with the average rank measure. Although the results in Figure 2 are already presented in Figure 1, the re-presentation shows differences that cannot be seen in in the ranking data in Figure 1.

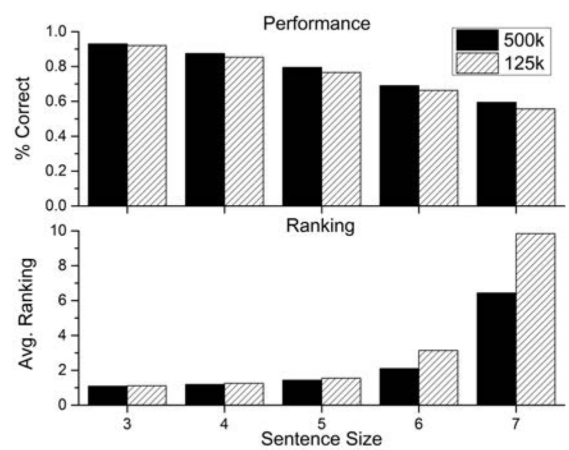

Figure 2. Performance from Figure 1 where m = 125,000 and results from a new simulation were m = 500,000

As shown, the mean rank was less than 2 for the three, four, and five word sentences, was less than 4 for the six word sentences, and was less than 10 for the seven word sentences. Given the average expected ranks for guessing with 3, 4, 5, 6, and 7 word sentences are equal to 3, 12, 60, 360, 2520, and 5040 respectively, the results are very impressive indeed. Plus, as we have already mentioned, whereas a rank greater than one indicates that the model failed to reproduce the exact order of words from the input sentence, the model still might have chosen an alternative syntactic completion—a possibility that increases with the number of words in the sentence. We plan to assess this possibility in the future.

Finally, the solid bars in Figure 2 show results for simulations where *m* was increased from 125,000 to 500,000—a larger sample of language experience that is more in line with work in other corpus based model analyses (e.g., Landauer & Dumais, 1997).

As shown, increasing the number of sentences in memory from 125,000 to 500,000 produced a modest 2-3% overall improvement of performance but with almost all of that improvement in judgments about the seven word sentences. This result reinforces our previous conclusion that after an initial 50,000 are stored in memory (see Figure 1), each additional sentence has a diminished impact on the model's performance.

In summary, Figures 1 and 2 represent a very high level of performance, even at large sentence sizes. For smaller sentence sizes of 3, 4, and 5, the model operated at greater than 75% correct, and was greater than 50% correct even at seven word sentences. Interestingly, much of the improvement in the model's performance was made with a relatively small number of sentences, with small improvements after 25,000 sentences. The reduction in performance across sentence sizes is linear, suggesting that as more sentence types are possible, the introduction of noise reduces model performance by a constant. In fact, the final ranking across the different sentence sizes was almost entirely due to the number of alternatives in the construction process, with a Pearson correlation equal to 0.99 between final ranking and number of permutations. As already discussed, this may also be a function of there being more possible syntactic constructions for words being greater with a higher number of words.

In conclusion, the model demonstrates a simple point: a raw record of language experience combined with parallel retrieval can provide a powerful mechanism to account for how people order words in sentences. The analysis also suggests that the body of linguistic experience need not even be overwhelmingly large and that a few (i.e., 50,000 exemplars) can go a long way to helping a person produce syntactic word orders in their natural language. Finally, the analysis also demonstrates that an appreciation of syntax can emerge from the application of a simple parallel retrieval mechanism applied to a realistically scaled representation of language experience.

General Discussion

Natural languages are defined by productivity and regularity. They are capable of producing an infinite number of different utterances, with all the utterances having a consistent structure. To account for these differing aspects of language it has been proposed that a formal grammar is necessary.

A formal grammar is a top-down approach, which seeks to understand language processing from the connections between abstract categories. The approach taken here with the Exemplar Production Model is the opposite of this: use a model that only knows the structure of past utterances, and use that experience to construct a future utterance. That is, it is a bottom-up approach, where past experience controls future behavior. The EPM was designed to exploit the productivity and regularity of natural language, in order to determine the power of experience in producing grammatical utterances.

The EPM is a simple model that encodes pure location and linear n-gram information to encode an exemplar of a sentence. A classic retrieval operation, based off of MINERVA 2, is used to construct the likely ordering of a sentence. Every possible ordering of a sentence is tested, with the ordering that is most similar to the expected structure being produced. There is no higher-level processing integrated into the model, and so the behavior of the model is entirely experience-dependent. In that sense, the theory is perfectly continuous with our previous efforts to build an exemplar-based model of language learning and comprehension using the same mechanisms and ideas (see Johns & Jones, 2015). However, there are some differences in the details of the current and previous models that need to be resolved before a complete integration of the two is realized. We take the problem of that integration as a challenge that would move toward the kind of model needed to generate a complete picture of how an exemplar-based model of memory can serve as a valuable competitor in the discipline's pursuit of a theory of language and language use.

The model was shown to be able to construct the correct ordering of simple sentences of sizes 3 to 7 to a high degree, with a linear drop in performance as sentence size increases. This demonstrates that past experience with language provides a large amount of power in producing grammatically correct utterances.

However, the really interesting part of the model's behavior is the performance of the EPM as a function of the number of exemplars it has studied. Performance rapidly increases with the first 20-25,000 sentences studied, with small improvements subsequently (the learning function most resembles a power law). Even when the total number of exemplars studied is quadrupled from 125k to 500k sentences, only a small improvement in performance is seen. However, it does provide a look into what the regular nature of language provides productivity: even with a small amount of linguistic experience, the correct structure of language emerges, due to the highly structured nature of natural language. Language is far from random and this consistency provides a simple model the ability to construct grammatically correct sentences, without any higher-level processing. As more exemplars are stored, the overlap in structure of the sentences emerges (due to the productivity of language), which allows for the model to exploit the combinatorial nature of language usage.

This is not to say that this approach does not have any challenges. The main one being that the model potentially operates at the wrong level of analysis – phrases may be the right unit of exemplars rather than whole sentences, as is the typical case in generative linguistics. Sentences then can be constructed by determining the correct order of phrases, integrating higher-level information into the exemplar construction process. This would also allow for the model to operate with lower number of words, which would be advantageous due to the model becoming computationally burdensome at a high number of words.

Another issue with the model concerns its encoding scheme: if it is generating the structure of a sentence of size *n*, it studies only exemplars of the same size. More research is required to determine the best mechanism to encode location in a relative fashion, where exemplars of differing length can be included in the same retrieval process.

However, these problems arise because of the simplicity of the approach, which is also its most promising feature. There is very little built into the machinery of the model and it still operates at a high level of performance. It provides a promising framework to examine language production and comprehension from a bottom-up point of view and allows for an examination into the power of experience in explaining linguistic behavior.

References

Abbot-Smith, K., & Tomasello, M. (2006). Exemplar-learning and schematization in a usage-based account of syntactic acquisition. *The Linguistic Review*, 23, 275-290.

Chomsky, N. (1988). Language and the problems of knowledge. *The Managua Lectures*. Cambridge, MA: MIT Press.

Christiansen, M., & Chater, N. (2008). Language as shaped by the brain. *Behavioral and Brain Sciences*, 31, 489-558.

Chubala, C. M., & Jamieson, R. K. (2013). Recoding and representation in artificial grammar learning. *Behavior Research Methods*, 45, 470-479.

Evans, N., & Levinson, S. C. (2009). The myth of language universals: Language diversity and its importance for cognitive science. *Behavioral and Brain Sciences*, 32, 429-492.

Hintzman, D. L. (1986). "Schema abstraction" in a multiple-trace memory model. *Psychological Review*, 93, 411-428.

Jamieson, R. K., & Mewhort, D. J. K. (2009). Applying an exemplar model to the artificial-grammar task: Inferring grammaticality from similarity. *Quarterly Journal of Experimental Psychology*, 62, 550-575.

Jamieson, R. K., & Mewhort, D. J. K. (2010). Applying an exemplar model to the artificial-grammar task: String-completion and performance for individual items. *Quarterly Journal of Experimental Psychology*, 63, 1014-1039.

Jamieson, R. K., & Mewhort, D. J. K. (2011). Grammaticality is inferred from global similarity: A reply to Kinder (2010). *Quarterly Journal of Experimental Psychology*, 64, 209-216.

Johns, B. T., & Jones, M. N. (2015). Generating structure from experience: A retrieval-based model of language processing. *Canadian Journal of Experimental Psychology*, 69, 233-251.

Jones, M. N., & Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114, 1-37.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104, 211-240.

McAuley, J., and Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems* (RecSys), 165–172.

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6, 623–641.

Reber, A. S. (1967). Implicit learning of synthetic languages: The role of instructional set. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 2, 88-94.