Production without Rules: Using an Instance Memory Model to Exploit Structure in Natural Language

Brendan T. Johns[1], Randall K. Jamieson[2], Matthew J. C. Crump[3], Michael N. Jones[4], & the late D. J. K. Mewhort[5]

[1]McGill University

[2]University of Manitoba

[3]Brooklyn College

[4]Indiana University

[5]Queen's University

In Press, *Journal of Memory and Language*

Corresponding Address
Brendan Johns
Department of Psychology, McGill University
2001 McGill College Avenue
Montreal, Quebec, Canada
H3A 1G1
Email: brendan.johns@mcgill.ca

Abstract

Recent research in the artificial grammar learning literature has shown that a simple instance model of memory can account for a wide variety of artificial grammar results (Jamieson & Mewhort, 2009, 2010, 2011), indicating that language processing may have more in common with episodic memory than previously thought. These results have been used to develop new instance models of natural language processing, including a model of sentence comprehension (Johns & Jones, 2015) and semantic memory (Jamieson, Avery, Johns, & Jones, 2018). The foundations of the models lie in the storage and retrieval of episodic traces of linguistic experience. The current research extends the idea to account for natural language sentence production. We show that the structure of language itself provides sufficient information to generate syntactically correct sentences, even with no higher-level information (such as knowledge of grammatical classes) available to the model. Additionally, we demonstrate that the model can account for a variety of effects from the structural priming literature (e.g., Bock, 1986). This work provides insight into the highly structured nature of natural language, and how instance memory models can be a powerful model type to exploit this structure. Additionally, it demonstrates the utility of using the formalisms developed in episodic memory research to understand performance in other domains, such as in language processing.

Keywords: Instance memory; Language production; Corpus-based modeling; Big data; Machine learning

Production without Rules: Using an Instance Memory Model to Exploit Structure in Natural Language

Human languages are both productive and regular. By productive, we mean that an infinite number of utterances are possible for any language. By regular, we mean that the utterances are systematically ordered. To explain these aspects of language, it has been proposed that language performance reflects the use of a formal grammar, due to the fact that a grammar of sufficient complexity can construct utterances of any length while maintaining consistency in utterance construction (e.g., Chomsky, 1957, 1988).

The idea that sentence production reflects the use of a formal grammar has proven powerful, with a great deal of linguistic intuition supporting its development (Robins, 2013). Grammatical accounts of language processing usually assume that an individual's experience with language does not include sufficient information to support linguistic competence. The notion underlying this position is that the human ability to use language outpaces what their capabilities should be, given the seemingly limited information available from the experience that people have with language (Gold, 1967). The position is often called "the poverty of the stimulus argument" (see Berwick, Poetroski, Yankama, & Chomsky, 2011; Laurence & Margolis, 2001; Perfors, Tenenbaum, & Regeir, 2006).

Analogous arguments favouring specialized processes with abstracted, higher-level representations have been advanced in many other areas in the cognitive sciences, most prominently in categorization (Rosch, 1973; Rosch, Mervis, Gray, Johnson, & Boyes-Braem, 1976). At the same time, alternative frameworks to abstractionist theories of categorization have been explored, mainly based around the storage of individual experiences, or instances (Brooks, 1978).

Instance-based theories propose that the individual experiences that people have with different aspects of their environment serves as the unit of knowledge on which cognition is based. For example, in the case of categorization, a person's ability to label a *dog* as a *dog* instead of a *cat* is based upon experiences the person has had with individual dogs and cats (e.g. Nosofsky, 1986, 1991). That is, the similarity structure in the known instances of a natural category provides sufficient information to support the required discrimination. Similar theories have been used in a number of other domains including automatization (Logan, 1988), recognition memory (Hintzman, 1988), attention (Krushke, 1992), word recognition (Goldinger, 1998), and schema abstraction (Hintzman, 1986; for an integrated review see Logan, 2002).

Artificial grammar learning is a second domain where abstractionist and instance models have been pitted against one another. Classic accounts of artificial grammar learning propose that when subjects study strings of symbols generated from a pre-defined grammar, they use the study experience to generate a representation of the grammar that had been used to construct the stimulus strings, and, in turn, use the newly acquired grammatical representation to discriminate grammatical from ungrammatical stimulus strings (Reber, 1967). Paradoxically, however, people are unable to articulate the rules that they used to accomplish the discrimination (i.e., their knowledge is implicit). Jamieson and Mewhort (2009, 2010), however, call such abstractionist explanations into question.

Jamieson and Mewhort demonstrated that a simple exemplar-based memory model (MINERVA 2; see Hintzman, 1986) can explain many of the relevant findings in the artificial grammar literature. The model works because it stores a record of the environment displayed during study (e.g., each presented string is stored as an episodic memory trace), and the stored instances contain information about the structure of the underlying grammar. As a result, decisions about the grammaticality of a particular string can be made on the basis of similarity to stored examples without assuming the need for high-level abstraction of the grammar.

More specifically, to explain a variety of phenomena from artificial-grammar tasks, Jamieson and Mewhort (2011) implemented storage and retrieval operations from a classic model of episodic memory (Minerva 2; Hintzman, 1986), combined with an account of word meaning taken from Jones and Mewhort's (2007) BEAGLE model, a computational theory of lexical semantics; which, in turn, is based in formalisms developed in the classic TODAM memory models of Murdock (1982; 1983). According to the account, each studied grammatical sentence is stored in memory. When a probe is presented at test, it retrieves all of the stored items 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 artificial-grammar learning and implicit memory 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, (d) variation in people's judgements depending on how they represent strings in memory, and (e) implicit judgements on high level category characteristics (Jamieson & Mewhort, 2009, 2010, 2011; Chubala & Jamieson, 2013; Chubala, Johns, Jamieson, & Mewhort, 2016).

The power of this model comes from the natural correlation between the form and amount of structure in an instance produced from a grammar, and the structure in the grammar itself (Jamieson & Mewhort, 2005). It follows, then, that each studied grammatical instance provides information about the underlying grammar. It also follows that a collection of grammatical instances will almost always provide a sum of information greater than that provided by one instance alone, where features shared in the items that are forced by structure in the grammar come to the fore and features that are idiosyncratic or unusual recede into the background. That is, it is the combination of instances that gives the model its power and it is the sum of instances that emerges in the act of parallel retrieval. The question, now, is whether the mechanisms of instance models, explored by Jamieson and Mewhort (2009, 2010) and established across a wide range of cognitions (e.g., Logan, 2002), can be extended to include natural language processing.

The use of instance technology to explain language processing is not without precedent; indeed, the approach has been championed by the usage-based perspective of child language acquisition (Tomasello, 2003; Abbot-Smith & Tomasello, 2006; see also Thiessen & Pavlik, 2013 for a different instance approach to child language development) and is based on a good deal of evidence that language development is item-based rather than acquired by abstracted representations over syntactic categories (Tomasello, 2000; Bannard, Lieven, & Tomasello, 2009). Instance models are also coherent with recent approaches to the cultural evolution of language (e.g. Christiansen & Chater, 2008, 2016) which propose that language processing evolved on top of domain general cognitive processes, such as memory. Additionally, they fulfil many of the requirements of recent general theories of language processing (e.g. Beckner, et al., 2009) that propose language is fundamentally adaptive in nature. A language that is adaptive entails that different language users should communicate in similar ways as other users of that language, given common social and cultural experience. Instance theories are adaptive as their behavior are completely dependent on past experience.

Related to the use of instance models as general models of language processing, there have been many developments in the computational modeling of semantic memory and knowledge acquisition that suggest recordings of environmental regularities provide a powerful basis for models of lexical behavior. Termed distributional models of semantics, these models have been used to assess the poverty of the stimulus argument in semantics (see Landauer & Dumais, 1997). This class of model learns the meaning of words through the analysis of large text sources (e.g. Griffiths, Steyvers, & Tenenbaum, 2007; Jamieson et al., 2018; Johns, Mewhort, & Jones, 2019; Jones & Mewhort, 2007; Landauer & Dumais, 1997; Mikolov, et al., 2013; for a review, see Jones, Wilits, & Dennis, 2015). Distributional models have demonstrated that there is an explicit connection in the representations that people have about the meaning of words and how those words are used in the natural language environment, similar to classic notions developed in linguistics and the philosophy of language (e.g., Firth, 1957; Wittgenstein, 1953). That is, distributional models challenge the orthodox view of innate knowledge (e.g., Chomsky, 1991; Quine, 1962; Shepard, 1987), and suggest instead that the environment provides sufficient information to acquire a sophisticated comprehension of word meanings. In other words, the richness of experience is the antidote to the poverty of a stimulus.

As a first demonstration that instance models of memory can provide a coherent account of language processing, Johns and Jones (2015) combined a standard instance memory retrieval process (MINERVA 2; Hintzman, 1986) with the learning mechanisms of a distributional semantic model, namely the BEAGLE model (Jones & Mewhort, 2007; Recchia, et al., 2015). Their model will be referred to as the Instance Comprehension Model (ICM). This model is part of a larger research effort attempting to integrate distributional semantic models with realistic cognitive processing models (see also Johns, Jones, & Mewhort, 2012; Johns, Mewhort, & Jones, 2017; Jones, Johns, & Recchia, 2012; Mewhort, Shabahang, & Franklin, 2018; Osth, Shabahang, Mewhort, & Heathcote, 2020).

BEAGLE is a specific type of distributional model entitled a vector accumulation model that encodes both the sentential context (the words that a specific word appears with in a sentence) and order (the relative position of words in a sentence) information. The ICM uses BEAGLE's encoding schemes to form sentence representations, and stores them as traces in an instance memory. By combining this encoding scheme with the retrieval processes used in MINERVA, Johns and Jones (2015) demonstrated that an instance model of language could account for a variety of findings, including sentence reading-time effects, integration of linguistic and perceptual processing, and the cultural evolution of language (see also, Johns, Jones, & Mewhort, 2019, who provide a mechanism to optimize this model type and extend it other types of data). The model's success provides a promising basis for establishing instance models as a coherent theoretical account of language comprehension. Additionally, it demonstrated that syntactic processing can be captured in a system that has no higher-level information integrated into its processing and representational architecture.

Jamieson, Avery, Johns, and Jones (2018; see also Kwantes, 2005 for an earlier approach and Crump, Jamieson, Johns, & Jones, 2020 for a refinement of Jamieson, et al., 2018) recently extended this approach by demonstrating that an instance model of memory can serve as a general model of semantic memory. The key insight of this model, and other related approaches (see Jones, 2018 for a review), is that the meaning of a word can be constructed at retrieval rather than each word having a singular vector representation, as is common in most distributional models. This insight allows the model to capture effects of polysemy on word meanings, an aspect of lexical semantics that most distributional models struggle with. Additionally, Kwantes and Mewhort (1999) proposed an instance model of lexical decision, demonstrating the utility of instance models to account for language-based behaviors.

Combined, the works of Johns and Jones (2015) and Jamieson, et al. (2018) show that an instance memory model can account for language comprehension data at both the single word and sentence level. Additionally, these models are scaled to have a similar level of experience that adult human beings have, demonstrating the flexibility of the approach. However, for instance memory models to be considered a viable model of language processing, it needs to be demonstrated that it can be extended to other data types.

Instance models are not the only model type that have been successful at capturing artificial grammar learning, as simple recurrent networks (SRN; Elman, 1990) have been shown highly capable of explaining a variety of results within this area (e.g., Cleeremans & McClelland, 1991; Dienes, Altmann, & Gao, 1999; Kinder, 2000; Kinder & Lotz, 2009; Servan-Schreiber, Cleeremans, & McClelland, 2001). Indeed, the arguments for and against these approaches have been discussed in the recent literature (see Kinder, 2010; Jamieson & Mewhort, 2011).

Even though both instance models and SRNs have been successful in accounting for results in the artificial grammar literature, they have very different theoretical bases. SRNs embody the original conceptualizations of Reber's (1967) notion of implicit learning, where it is assumed that the rules of the grammar are being abstracted across repeated experience with the strings of a grammar, and those abstracted rules are contained in the connection weights of the neural network. As stated previously, instance models provide a much different explanation of performance on this task, where it is proposed that there is no abstraction across experience during learning in an artificial grammar task. Instead, the ability of the model to discriminate grammaticality lies in the similarity of a test string to previously experienced strings (Brooks, 1978; Brooks & Vokey, 1991; Jamieson & Mewhort, 2010, 2011), or abstraction at retrieval.

One area where SRN models have been successfully applied is in language production (Chang, Dell, & Bock, 2006). The SRN model of Chang, et al. (2006) is able to account for many classic and contemporary results in language production, particularly results in structural priming (Bock, 1986; see Pickering & Ferreira, 2008 for a review). In language production, structural priming is the finding that the syntactic structure of previously processed sentences impacts the syntactic construction chosen for a future production. Chang, et al. (2006) used an SRN as the basis of a model which was capable of learning word sequence information, as well as form-meaning mappings, in order to explain a large variety of results from the language production literature (see Reitter, Keller, & Moore, 2011 for a different approach to account for this data using an ACT-R architecture).

The goal of this article is not to provide a complete explanation of language production, as was the goal of Chang, et al. (2006). Given that the majority of data in the language production literature use picture production tasks, a model requires a perceptual processing module (or assumptions about how perceptual processing operates). Although Johns and Jones (2015) demonstrated that an instance model of language can integrate perceptual information into its comprehension processes, that is outside of the goals of this paper. Instead, the question that this article is attempting to answer is the power that an instance model provides as the underlying basis for a language production model, and combined with the results of Johns and Jones (2015) and Jamieson, et al. (2018), an integrated model of language comprehension and production.

The overall goal of the current article is to extend the instance approach of cognition to language production, with the additional goal of demonstrating the power and flexibility of instance memory models (in combination with well-known distributional approaches to lexical semantics) in accounting for complex lexical phenomena. Instance models are attractive because the complexity of the model lies in the complexity of the information that has been encoded. We will estimate the power of the structure in samples of language by storing sentences within an instance memory and then assessing how effectively the stored information can generate grammatically correct utterances and also whether it can account for behavioral results from the structural priming literature. In particular, we will show that the productive and regular nature of language provides enough structure to produce grammatically correct utterances, even though the model will have no explicit grammatical knowledge. Instead, the model's successes will emerge simply from the formation of memory traces that encode the regular nature of language.

An additional goal of this article is to demonstrate how a standard computational model of memory can have useful properties when scaled up to realistic levels of human experience, integrating memory modeling into new trends in big data approaches to cognition (see Jones, 2017; Johns, Jamieson, & Jones, 2020). In a typical computational model of memory, the model is given enough information to simulate a task- and stimulus-specific experimental setup. However, this ignores both the depth and breadth of lifelong learning (Qiu & Johns, 2020). By analyzing how memory models respond to a large number of traces, it provides insight into the dynamics of memory retrieval at a larger scale. It further suggests that the mechanisms that humans use for language production may have been exapted from mechanisms that we originally evolved for episodic memory retrieval, coherent with the proposals of Christiansen and Chater (2008). The results of this article will point to instance memory models being a promising model type to examine the effects of accumulated experience on behavior.

However, we do not wish to propose that the model outlined below is a complete explanation of language production, or even close to be so, as it is lacking many central cognitive and language processing aspects that would allow for it to be considered so. Instead, we wish to demonstrate that an instance approach to language production contains some of the necessary components to account for natural language production data, while making minimal assumptions about the nature of linguistic representation in the mind. Thus, this work will serve as an existence proof of the power of an instance language model to account for language production behavior, serving as a complementary piece to the development of instance models of language comprehension (Jamieson, et al., 2018; Johns & Jones, 2015; Jones, 2018), and also demonstrates the generality of the instance approach to various forms of cognition.

The first section of this paper will provide an overview of the mechanisms of our modeling approach. The second section of the paper will demonstrate the capabilities of the model, by showing that the model is capable of constructing grammatical sentences from unordered sets of words under various manipulations. The third section contains simulations demonstrating that the modeling framework developed can account for various standard effects from the structural priming literature.

The Instance Production Model (IPM)

The IPM model will have a similar conceptual basis to the previously discussed ICM model of Johns and Jones (2015), with some modifications to how memory traces are formed.

The ICM combined a MINERVA-like retrieval process with the learning mechanisms of the BEAGLE model of semantics (Jones & Mewhort, 2007; Recchia, et al., 2015). In vector accumulation models, such as BEAGLE, words are initially represented by randomly generated static environmental vectors, which are assumed to represent the perceptual properties of a word. To encode the location of a word within a sentence, the ICM used the random permutation techniques developed by Sahlgren (2006) and Recchia, et al. (2015). The models of Sahlgren (2006) and Recchia, et al. (2015) utilize binary splatter codes (sparse ternary vectors) for environmental vectors, where non-zero values are either +1 or -1 with equal probability. These vectors are typically very sparse. Random permutations simply randomly shuffle the elements of a word's environmental vector. Each location within the sentence is given a unique permutation. A word in a location uses that random permutation to form a unique binding between that word and that location within a sentence. The representation of sentence is then the sum of all the permuted environmental vectors into a single composite vector. The successes of Sahlgren (2006), Recchia, et al. (2015), and Johns and Jones (2015) demonstrate the power of absolute order in language: the serial positions of words in sentences provide a great deal of information about the usage and thus the meaning of that word.

In contrast to Sahlgren (2006) and Recchia, et al.'s (2015) techniques, the original BEAGLE model was built using circular convolution (Kelly, Blostein, & Mewhort, 2013; Plate, 1995) to encode n-gram information. Circular convolution is a function that takes in two vectors and constructs a new, unique vector that represents the association between the two. In

BEAGLE, this results in a sophisticated representation of the order that words are used in a language, constructed through exposure to natural language sentences in a large text base.

Technically, BEAGLE utilizes non-commutative circular convolution to encode serial-order information, such as encoding linear order n-grams within a sentence. Non-commutative circular convolution is accomplished by scrambling indices in a word's environmental vector differently depending on whether it is the predecessor or successor in a bigram (see Jones & Mewhort, 2007; Plate, 1995).

The important aspect to understand about circular convolution for the purposes of this article is that it allows for a unique vector representation of chunks of language, such as bigrams and trigrams (although larger chunks are possible; Jones & Mewhort, 2007), to be formed. For example, consider the sentence "the girl ran home." Using linear bigram and trigrams, unique vectors are formed for the following chunks: *the girl*, *girl ran*, *ran home, the girl ran,* and *girl ran home*. By collapsing these bigram and trigram vectors into a composite, a simple encoding of syntactic structure is formed. Importantly, chunks that are high in frequency (e.g., *the girl*) across experience provide substantial constraint about the correct ordering of sentences, as the following simulations will show. In the remainder of this paper, we denote non-commutative circular convolution with the $\circledast$ symbol (e.g., $\mathbf{z} = \mathbf{x} \circledast \mathbf{y}$). For brevity, we will use the term convolution in place of non-commutative circular convolution.

Thus, there are two main types of information that can be encoded within an instance for a sentence: absolute-position (pure location information) and relative position (n-gram information). Given that Recchia, et al.'s (2015) technique was not designed to encode n-grams, their representational schema is not applicable here.

Although it has not been done previously, both types of information can be readily captured within the BEAGLE framework. Instead of representing a location within a sentence with a random permutation, locations can be represented by a random vector (generated in the same fashion as an environmental vector). Then a word vector can be convolved with a location vector in order to generate a unique signature of a word occurring in that location. By doing this for each word in a sentence, and summing the resulting vectors into a composite vector, a linear ordering of a sentence is obtained.

The advantage to using this framework is that it allows for linear n-gram information to also be integrated into the representation, in accordance with the standard mechanisms of

BEAGLE (allowing smaller chunks of sentences to be integrated into the representation), along with pure location information. This combination allows the model to capitalize on the successes of both the ICM and the original BEAGLE model.

In addition to order information, BEAGLE also encodes context information (operationalized in terms of the other words that occur with a target word in a sentence). Context information is encoded as the sum of all of the environmental vectors (except the target word) that occur in a sentence. Sentential context and order information offer different knowledge about the usage of a word, and have been shown to account for complementary types of linguistic information (Jones & Mewhort, 2007; Hare, Jones, Thomson, Kelly, & McRae, 2009; Johns, et al., 2018). In memory, each sentence processed by the model will have both a context and order vector concatenated together.

Additionally, in the IPM context and order vectors will also have a different conceptual basis and purpose. The communicative intent behind a to-be-produced sentence is represented by its sentence context vector, which carries no order information. That is, a context vector represents what the model wants to produce. Successful sentence production requires that order be imposed onto the words in the sentence context vector. In IPM, order information is applied to a sentence context vector at the time of memory retrieval. The context vector is treated as a probe or retrieval cue (e.g., a set of unordered words), that retrieves similar sentences stored in memory. In particular, the context cue will be used to retrieve the likely ordering of a sentence, based upon the pattern of instances stored in memory. Stored sentences have both context and order information, and the retrieved order information from memory is applied to sequence the words in sentence context probe. In other words, given a set of unordered words, the goal of the IPM is to retrieve the correct syntactic ordering for those words. We now express the model more formally.

Representation. In the model, each word is represented by its own unique random environmental vector, **w**, of dimensionality *N*, where each dimension takes a randomly sampled value from a normal distribution with mean zero and variance $1/N$. In the simulations that follow, $N = 2{,}048$.

Each sentence is represented by two vectors, context and order, both of which are constructed from the environmental vectors.

The sentence's context vector, **c**, is computed as,

$$\mathbf{c} = \sum_{i=1}^{n} \mathbf{w}_i \quad (1)$$

where **c** is the context vector, n is the number of words in the sentence, and $\mathbf{w}_i$ is the environment 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.*"

A word's order vector is equal to the sum of location information and linear bi- and trigrams. The order vector, **o**, is computed as,

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

where **o** is the order vector, n is the number of words in the sentence, $\mathbf{w}_i$ is the word in serial position i, $\mathbf{w}_{i-1}$ is the word in serial position $i - 1$, $\mathbf{w}_{i-2}$ is the word in serial position $i - 2$, $\mathbf{l}_i$ 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 (c) 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. Additionally, in the following simulations, the power of the individual components will be tested both independently and jointly to identify the most parsimonious combination of lexical sources.

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 1,000 sentences is represented in a 1000×4096 (both the context and order vectors have 2048 values) matrix whereas memory for 125,000 sentences is represented by a $125{,}000 \times 4096$ matrix.

Retrieval. Retrieval in the model is parallel, probe-specific, and similarity driven. 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 the whole traces record both context and order information in the sentence, a probe that includes just the context information also retrieves the order information that it has co-occurred with in the past. This is how the model simulates cued-recall; 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 first step of the retrieval process is to activate each of *i* traces according to the similarity between the probe **p** and the memory trace $\mathbf{M}_i$:

$$A = S(\mathbf{p}, \mathbf{M}_i)^{\lambda} \quad (3)$$

where **p** is the context vector that encodes an unordered list of words (i.e., includes information in serial positions 1…*N* with serial positions *N*+1…2*N* set to zero), **M** is the memory matrix that stores the model's sentence knowledge, *i* is the location of the trace being activated in memory, and λ is a fixed scaling parameter that controls the impact of any single trace on memory retrieval (this parameter was set at 9 in the following simulations to simulate a very selective trace specific retrieval). The similarity function used is a vector cosine, and is calculated as follows:

$$S(\mathbf{x}, \mathbf{y}) = \frac{\sum_{j=1}^{N} \mathbf{x}_j \times \mathbf{y}_{ij}}{\sqrt{\sum_{j=1}^{N} \mathbf{x}_j^2}\sqrt{\sum_{j=1}^{N} \mathbf{y}_{ij}^2}} \quad (4)$$

where N is the size of the vector. Then, the echo, **e**, is computed as,

$$\mathbf{e} = \sum_{i=1}^{m} A \times \mathbf{M}_i \quad (5)$$

where **e** is the retrieved echo, and *m* is the number of sentences stored in memory. As with a sentence representation, features 1…*N* in **e** represent the context vector retrieved from memory and features *N*+1…2*N* in **e** represents the order vector retrieved from memory.

Decision. Our goal is to measure the model's ability to produce a syntactically correct sentence composed of words presented to the model in an unordered word 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 of 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, $\mathbf{c} = \mathbf{w}_{eat} + \mathbf{w}_{\text{your}} + \mathbf{w}_{dinner}$, 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).

The benefit of this production mechanism lies in its simplicity: the chosen ordering of a sentence will be the one most coherent with the structure of past experience. There is no sophistication built into the production mechanism as it allows for a baseline to be established about the power of experience in constructing grammatically correct sentences.

Methods. 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, where *n* will range between 3 and 7 word sentences. Second, we computed the model's ability to translate each of 200 unordered word lists of length *n* into ordered sentences of length *n* (that is, to take a set of unordered words and construct a syntactically correct sentence).

We expect the model will re-write unordered word lists as syntactically correct sentences. That is, we expect the model to take a list of unordered words as a memory cue and use its past experience with natural language to order those words in a sound syntactic manner. If true, our simulations would demonstrate that parallel retrieval from a record of language is sufficient to produce a behavioural hallmark of syntactic behavior and would add to the growing literature emphasizing the importance of an individual's experience with language as well as the use-based approach to language learning and language theory (Abbot-Smith & Tomasello, 2006; Jamieson & Mewhort, 2010, 2011; Johns & Jones, 2015). See Chang, Lieven, and Tomasello (2008) for a similar model evaluation scheme.

Test Sentences. We assembled a pool of 30,000,000 sentences from a number of sources including Wikipedia articles, Amazon product descriptions (attained from McAuley & Leskovec, 2013), 1000 fiction books, 1050 non-fiction books, and 1500 young adult books. The properties of these corpora can be found in Johns, Jones, and Mewhort (2019). Once collated, 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 test-sentence occurrences were removed from the training material, so that the model would have never experienced a test sentence before, meaning that the model is being asked to produce sentences that has not previously seen.

All sentences are simple in construction, and use mostly high frequency words with personal pronouns, but, given the complexity of the task, they should provide a useful assessment of the model's performance.[1] These sentences have been used by others for model validation as well (see Kelly, Mewhort, & West, 2017; Kelly, Reitter, & West, 2017). No punctuation was included in the representations of the sentences. No general syntactic construction was used, but the majority consist of single phrase structures. Sample test sentences can be found in Table 1.

Simulation parameters

The simulations focused on two key parameters: sentence length (i.e., n) and language experience (i.e., m). Sentence length was assessed by conducting separate simulations for sentences of length x = 3, 4, 5, 6, and 7. That is, for simulations involving constructing three word sentences, the memory matrix only contains representations of three word sentences (see the General Discussion for an examination of this issue). As sentence length increases so does the combinatorial problem that the model faces. For sentences of length 3, 4, 5, 6, and 7, there are 6, 24, 120, 720, and 5,040 possible orderings, respectively. Only sentences of a set length were included in a simulation (i.e., if sentences of length 5 were being tested, the model was only trained with sentences of length 5). Language experience was assessed by conducting separate simulations given m = 1000, 2000, 3000 … 1,000,000 sentences stored in memory. Average performance was assessed by resampling the sentences selected 25 times to ensure that the results were not conditional on a particular record of language experience.

Although it is difficult to map model experience onto developmental trajectories, the corpus size used to train the model is likely a fraction of what a typical adolescent or adult would

[1] All test sentences can be found at http://btjohns.com/experience_sents.zip.

have experienced. Studies indicate that a typical student should experience around 12 million word tokens a year just from spoken language alone (Brysbaert, Stevens, Mandera, & Keuleers, 2016; Mehl, Vazire, Ramírez-Esparza, Slatcher, & Pennebaker, 2007). Given an average sentence size of roughly 13 words per sentence (Johns, et al., 2019), this gives a conservative estimate that people hear roughly a million sentences a year. The maximum number of training used in the below simulations is a million sentences, so corpus size is well within realistic levels of experience.

Two measurements of performance

We measured sentence-completion performance two ways. The first method tallied the percentage of tests in which the model's most preferred word order corresponded to the original sentence. The second 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, recorded 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 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 measure offers a more nuanced assessment: it 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.

Discussion. The IPM provides a simple framework for examining the power of past experience in sentence production. It uses the BEAGLE framework (Jones & Mewhort, 2007; Johns & Jones, 2015; Recchia, et al., 2015) to generate instances of sentences. To retrieve the latent ordering of a set of words, its retrieval mechanism was inspired by Hintzmans's (1986) MINERVA 2, and it used Jamieson and Mewhort's (2009, 2010, 2011) method for explaining artificial grammar learning, Johns and Jones's (2015) technique for examining sentence comprehension, and Jamieson, et al.'s (2018) model of lexical semantics. Thus, it integrates a growing body of research on the importance of instance memory on language processing. To generate the ordering of a set of words, every possible ordering is tested by its similarity to the latent ordering, with the most similar being what is produced.

The goal of the first set of simulations presented below is to determine the IPM's capability at producing grammatically correct sentences, given no higher-level information. The model depends almost completely on experience. The ordering produced for a sentence is based on the ordering seen in the past. If the model is capable of imposing structure on a set of words at a highly accurate degree, it follows that experience with language combined with similarity based and parallel retrieval of that experience is sufficient to explain at least some aspects of language usage independent of higher-order grammatical information. The goal of the second set of simulations is to demonstrate that the model can also account for standard behavioral effects of structural priming on language production. Overall, the goal of the combined simulations is to provide an existence proof that an instance model can account for aspects of complex linguistic phenomena without the need for higher level grammatical knowledge to be built into the model.

Production Simulations

Ordering test sequences. Before examining the model's overall performance across different sentence lengths, we first determined the optimal and most parsimonious combination of lexical features to form memory instances. Six different combinations of lexical features were tested – location, bigram, and trigram bindings by themselves, plus location + bigram, location + trigram, and location + bigram + trigram. The model was tested on sentences of five words, and was trained with 1,000,000 sentences. For five-word sentences, there are 120 possible combinations for the model to discriminate among, making this a non-trivial test of the model's discriminative capabilities.

The results are displayed in Figure 1 for both correct production (i.e., how often the sentence produced the correct ordering) and ranking (i.e., the rank of the correct sentence in the distribution of all sentences, with zero being the sentence ordering that was selected) performance.

As shown in Figure 1, location binding is the best performing information source, suggesting that where a word occurs in a sentence is an important source of syntactic information. The result is consistent with past results from distributional semantic modeling (Sahlgren, 2006; Recchia, et al., 2015), and is important because it can be easily encoded within the BEAGLE framework. In addition, note that the model's best used location + bigram information for both the production and ranking assessments; the inclusion of trigram information did not provide an increase in performance. In the following simulations, the location + bigram model will be used to form sentence instances, given that it is the simplest and best performing combination of lexical sources.

As discussed previously, the IPM's production mechanism searches through each possible combination of words in order to choose the most likely ordering, given the model's past experience. This leads to a combinatorial problem. For sentence lengths 3, 4, 5, 6, and 7 there are a respective 6, 24, 120, 720, and 5,040 combinations of the words in those sentences. Thus, the scale of the search problem grows with sentence length, providing an increasingly discriminative test of model's performance.

Figure 2 shows both the production performance and ranking measures as a function of sentence length. The model generated the correct ordering of a sentence at a high rate of success at all sentence lengths, ranging from 92% correct (chance = 16.7% correct) for three-word sentences to 76% correct (chance = 0.02% correct) for seven-word sentences. The correct-ordering data demonstrate that the model does have to ability to construct sentence ordering, even without any higher-level grammatical information built into the model. The ranking data shows the ability more directly: the average ranking of a sentence is directly correlated with the number of possible combinations of a sentence. Given that the model is able to generate the correct ordering of a sentence at such a high rate of success across sentence length, the ranking data demonstrate that an instance model can serve as a powerful basis for understanding sentence production.

A second important consideration for the IPM's performance is the model's performance across learning. Hence, the model's performance across sentence lengths was assessed every one thousand sentences, up to the limit of one million sentences studied. The results for both the performance and ranking data are contained in Figure 3.

Figure 3 shows that the vast majority of the model's increase in performance happens with relatively little linguistic input (i.e., less than fifty thousand sentences), with small increases occurring after this period. The point is especially clear in the ranking assessment. It follows that linguistic input must be structured so that a relatively sparse sample of language can support highly syntactically structured language production, with small changes occurring later in learning (especially for longer sentence lengths).

It is worth noting that the model's performance may be greater than is reported here. Some sentences have several equally valid syntactic constructions. For example, the model preferred "they quietly went down the stairs" when tested on "they went quietly down the stairs." Although the model did not produce the input sentence, it nevertheless generated a syntactically valid alternative.

To better understand model behavior across learning, consider Table 2 that shows the model's output across learning for seven different six-word sentences. As shown, there is variability in how fast the model learns the correct ordering. For example, the model produced the correct ordering of the sentence "you poked him in the eye" after encoding 28,000 instances to memory but needed 336,000 instances to produce the correct ordering of "the man was suffering from depression." This likely has to do with the frequency of the occurrence of words in the database and thus memory. Table 2 also shows that the model can produce sentences that are grammatically correct (or nearly correct). For example, given less language experience, the model preferred "the boy was a good dog" and "the man from depression was suffering" before hitting on the correct ordering of those sentences.

Figure 4 shows a quantitative examination of model performance across instance learning that plots the average similarity of the different sentence constructions for the correct ordering and the possible alternatives for sentences of four words (four word sentences were chosen to aid visualization) given 1,000 sentences are encoded to memory (top panel) versus 20,000 sentences encoded to memory (bottom panel). As shown, as the model encodes a greater number of instances, the correct ordering becomes increasingly more activated relative to the alternatives. However, even when a mere 1,000 sentences are encoded to memory, the correct ordering has a greater level of activation; a result that suggests the redundancy of language emerges early in the recording of experience.

Comparison to alternative base-rate model. One way to put the results into context is to compare the model's performance to a standard statistical language model, a bigram model. In a bigram model, the probability of producing word W_n when preceded by word W_{n-1}, is calculated as:

$$P(W_n|W_{n-1}) = \frac{P(W_n, W_{n-1})}{P(W_{n-1})} \tag{6}$$

That is, it is the probability of word W_n being produced following word W_{n-1}. To generate the most likely ordering of a set of words, we used the same procedure as the IPM: The different permutations of a set of words was generated, and the most probable ordering was selected by choosing the ordering that maximized the probability equation:

$$\Pr(W_1, W_2, \ldots, W_n) = \prod_{i=1}^{n} \Pr(W_{i+1}|W_i) \quad (7)$$

Put differently, the bigram probability of each ordering was calculated, and the most probable ordering was selected.

To assess the simple bigram model, its performance was compared against the IPM's performance and against the IPM with only bigram information encoded. The IPM bigram-only model was tested to assess what the additional information is contributed by the IPM's retrieval mechanism, over and above the direct statistics contained in the language source.

Figure 5 shows the results of this simulation for both production and ranking scores. As shown in Figure 5, both the IPM and IPM-bigram models significantly outperformed the standard bigram model, especially at longer sentence lengths. The IPM's ability to outperform the bigram model signals that the instances contained in memory contain more than just lexical statistics about the correct ordering of sentences. Particularly, the ability to efficiently encode locative and bigram information into a single instance representation allows for rich information about the likely ordering of a set of words to be retrieved. Additionally, the preferential activation of traces that contain some of the words in a to be produced utterance, but not all of the words, allows for some constraints on the ordering of a sentence. For example, if the model was asked to construct the sentence "the girl ran away" and the model had processed the sentence "the girl ran home" previously, the already processed sentence is not identical, but still provides considerable constraint to the likely ordering of a sentence. That is, the retrieval mechanism does not just return noisy bigram information; rather it includes extra-item information that allows for better order construction.

Filling in missing words. Thus far, we have shown that the IPM can order words syntactically based on past experience with language. But, can it also choose and use function words to stitch together a sentence composed of given content words? To test the possibility, we removed function words from test sentences and determined the accuracy with which the model filled-in the missing words (and the order of the words). The function words used were the 15 most frequent function words from the corpus.[2] Sentences of length 5 and 6 were taken from the previously used test sentences. All sentences that included one or two function words were retained. The model's task was to construct a sentence that included a set of content words and

[2] The set of function words used were: {the, to, of, and, a, in, I, that, you, for, it, on, this, with, be, have, as, or, if, but}.

do so by selecting and using missing function words. For example, if the model was given the test sentence “it smells like trouble to me” the resulting cue would contain the words “smell like trouble me” in an unordered list and it would need to both determine the missing function words and sew the content words together accordingly.

The fill-in manipulation increased the model’s combinatorial problem significantly, especially in the two-removed condition, as every possible combination of the missing function words has to be tested. In a five-word sentence, there are 15(5!) = 1,800 possible combinations in a one-removed condition and $15^2(5!) = 27{,}000$ combinations for a two-removed condition. For the six-word sentences, there are 15(6!) = 10,800 combinations in the one-removed condition, and a very substantial $15^2(6!) = 162{,}000$ combinations in the two-removed condition. Because the numbers of producible sentences are so large, the chance of producing the target sentence by accident is very small. Thus, the fill-in test provides a very discriminative assessment of the IPM’s ability to find structure in experience. In the following simulation, the function words were removed from the context vector, so that the memory probe did not contain any information about the missing words.

Figure 6 shows the results as a function of both the performance and ranking measurements. For the one-removed condition, performance was approximately equal to the performance of the model on the simpler ordering task used previously. That is, the IPM is eminently capable of not only ordering words but also generating the correct function word that completes a sentence.

The ranking data in this simulation did increase slightly compared to previous simulations, reflecting the greater number of possible combinations of words (this was also found in the two-removed condition). In the two-removed condition, there was a definite decline in the model’s performance (of approximately 7.5% for five-word sentences and 13% in the six-word sentences). Nevertheless, the level of performance is still quite high considering that the size of model’s search set. The model was able to generate both the correct two missing function words and ordering for sentences of six words at over 60% of the time, even though chance in this condition was only 0.00067%.

Generalizing to another language. Given that we are claiming that instance memory provides a domain general mechanism to account for language production, the IPM should be able to generalize to another language. To test the power of the model on another language, a corpus of novels in French language was used (first described in Jones, Dye, & Johns, 2017). The corpus contained 65 million words and is smaller than the English corpus used previously. Hence, in the simulation, the model encoded up to 500,000 sentences. Using the French corpus, 200 sentences of sentence lengths 3-7 were assembled.[3] The model was then run using the same parameters as in the previous simulations with the English corpus.

Figure 7 compares the results trained on the French corpus with the results using the English corpus. As shown in Figure 7, the results are remarkably similar for these sentence sets, with the English version performing slightly better for the shorter sentence lengths and the French model performing slightly better for the longer sentence lengths. The results suggest instance memory models are not tied to the statistical patterns of English but also extend to at least one other language. The demonstration points to universality.

Discussion. The above simulations demonstrate that an instance model, with very limited assumptions about the nature of linguistic experience built into the model, can take a set of words and generate the correct ordering for those words (across both English and French), as well as generating missing function words in a sentence. The success of the IPM points to the overall redundancy of the language environment, such that the utterances that people use are not randomly constructed, but instead contain significant overlap across instances. That is, even though many utterances contain somewhat different structures, when abstracted across thousands, tens or hundreds of thousands of sentences, an instance model is capable of retrieving the grammatically correct ordering of a sentence.

Even though the model is capable of this feat, it still does not reach maximum performance at larger sentence sizes (which nearly all speakers could almost always produce effortlessly). Additionally, the sentence set used to test the model are of very simple constructions. However, the above simulations offer an existence proof that an instance model does seem to provide a basis for the construction of grammatical sentences, even without any grammatical knowledge built into the model. The next simulation set will demonstrate that the model is capable of accounting for behavioral data from the structural priming literature.

Structural Priming Simulations

[3] Sentences can be found at http://btjohns.com/french_sents.zip. Due to none of the authors being completely fluent in French, sentences were attained by finding sentences that occurred three different times across the corpus, ensuring that the construction had been used by different authors.

Structural priming, as first discovered by Bock (1986), is the finding that the syntactic constructions of previously processed sentences impacts the syntactic construction chosen for a current utterance (for a review, see Pickering & Ferreira, 2008). For example, if a person had previously produced an active transitive sentence (e.g., "*A janitor cleans the floor daily*"), they are more likely to use that same construction for a future utterance (e.g., "*A brick struck the car's windshield*") rather than an alternative construction (e.g., "*The car's windshield was struck by a brick*"; a passive transitive sentence). Structural priming has served as an important development in the field of psycholinguistics (Dell & Ferreira, 2016) and has been found to be a powerful and consistent effect across many manipulations (Mahowald, James, Futrell, & Gibson, 2016).

In Bock (1986), subjects were made to produce sentences of a certain type (dative or transitive sentences) and were then given a picture description task, where the subject had to provide a short sentence describing the events taking place in the picture. It was found that subjects were more likely to use the syntactic construction of a prime sentence in the picture description task, compared to when the picture was preceded by a prime of a different syntactic construction. Importantly, the prime sentence had little thematic or meaning overlap to the target sentence, meaning that the phenomenon seems to rely upon previously used abstract representation of syntactic constructions in language production.

As stated previously, the ability to account for structural priming is a key evidence source for the use of SRNs in modeling language production (Chang, Dell, & Bock, 2006). The model of Chang, et al. (2006) is of significant complexity, and a full description of the model is out of the purview of the current article. However, the conceptualization of the model relies on the notion of syntactic adaptation to current language input, where across lexical experience the syntactic constructions being used in the environment are abstracted across time with a connectionist network using error-driven learning. Due to this adaptation, the model is more likely to utilize that syntactic construction in the future. It is proposed that this adaptation is a form of implicit learning (Bock & Griffin, 2000). Chang, et al. (2006) demonstrated that this approach could account for most of the major findings in the field.

The view of structural priming as abstracted implicit learning echoes previously outlined arguments in the artificial language literature (e.g., Kinder, 2010; Jamieson & Mewhort, 2011), contrasting implicit learning versus memory-based explanations. One of the main arguments in structural priming pointing to the use of implicit learning is the finding that structural priming occurs across significant time lags (Bock & Griffin, 2000; Bernolet, Collina, & Hartsuiker, 2016). These findings have been used to directly argue against the notion of a memory-based explanation of structural priming (Bock & Griffin, 2000).

The goal of the following simulations is not to give a complete accounting of structural priming in language production, but instead to show that an instance memory model, trained on a natural language corpus without any higher-level grammatical information being built into its representation or processing assumptions (in contrast to the approach of Chang, et al., 2006), can in principle account for some of the foundational results in structural priming. Additionally, we will show that the processing assumptions of an instance model of structural priming are coherent with the theorizing of usage-based (Abbot-Smith & Tomasello, 2006; Tomasello, 2001) and adaptive (Beckner, et al., 2009; Christiansen & Chater, 2008) approaches to language processing.

In the simulations reported so far, the basic operation of the IPM was to take a set of unordered words (as encoded in a context vector; see equation 1), and retrieve the likeliest ordering for those words through the cueing of memory, akin to a cued recall operation. The conceptualization of an instance model of language production lies in usage-based theories (Tomasello, 2001; Abbot-Smith & Tomasello, 2006): in order to produce utterances that are comprehensible to others in their environment, use a similar structure to the utterances that were produced by others. The simulations so far have demonstrated that this is a powerful approach to producing syntactically correct utterances.

However, it is difficult to see how structural priming could be accounted for within this modeling framework as currently constructed, given that the cue to retrieve syntactic structure (the context vector), by definition, has no structure embedded into it. As previously stated, the context vector should be considered as a communication intention, or the meaning of what is wanting to be expressed. Structural priming is a phenomenon demonstrating the local nature of language processing – the linguistic context that one is embedded in impacts how one produces language. That is, how one expresses themselves is not tied to internal motivations only, but also to information contained in the external environment. From a usage-based perspective, the important factors of the external environment are others that you are attempting to communicate with. If one knows how others communicate in a context, then one should attempt to communicate in similar ways.

Still, the question lies in how to adapt the IPM to local context. Order vectors, as constructed with equation (2), encode how a sentence was produced. If it is assumed that a short-term memory store holds order information about the utterances being produced in a context, then there is a natural mechanism by which local communicative context can be naturally built into the retrieval mechanism of the IPM: by cueing with both context and order vectors. If a context cue represents *what* is supposed to be produced, an order cue represents *how* it should be produced. By integrating an order cue into the retrieval operation, the retrieved structure of an utterance can be biased towards certain constructions by using an order cue. The integration of multiple cues in the retrieval process is consistent with other modeling efforts into syntactic priming (e.g., Dosher & Rosedale, 1997; Ratcliff & McKoon, 1994).

A central tenet of modern theories of language is that they are adaptive (Beckner, et al., 2009). Only cueing with a context vector means that the model is globally adaptive: the model produces language based on the overall language statistics that the model has been exposed to (see Dalton, 1993 for a discussion on global versus local contextual features). However, by integrating an order cue based on the current environment the model is embedded in, the model becomes locally adaptive: the model will be biased to produce utterances using a similar construction to the utterances that others are using in the current context. By this account, structural priming is not the result of retrieved abstract syntactic representations in the brain, but by a communicative need: to produce language like others in your current environment are using language, and thus maximizing communicative effectiveness, or maximizing linguistic adaptivity.

To integrate this retrieval mechanism into the IPM is simple: instead of only taking the similarity between the context cue and context memory trace (as was done in equation 3), take the average of the similarity between the context cue and context memory trace and the order cue and the order memory trace. The resulting activation of a trace in memory is thus a mixture of both the words that are trying to be produced and also the structure of previously processed sentences contained in short-term memory. The information contained in the echo is thus a reflection both of what is trying to be expressed and also how it should be said to optimize local comprehensibility.

In order to integrate an order cue into the retrieval process, the trace activation formula in equation (3) has to be modified:

$$A = \left(\frac{S(\mathbf{c},\mathbf{m}_i)+S(o,\mathbf{m}_i)}{2}\right)^{\lambda} \tag{8}$$

Where **c** is the context cue and **o** is the order cue, and $\mathbf{m}_i$ is the instance in memory. All other aspects of the retrieval process were kept constant with previous simulations.

Other retrieval cues could be used. For instance, the context and order cue could simply be concatenated to form a single cue. However, that would require normalization between differences in magnitude between context and order vectors. Multiplicative similarity could also be used to activate traces, but that would introduce different retrieval dynamics. Average similarity provides a simple mechanism which is reflective of the machinations used in the previous simulations of this article.

Simulation outline. Most structural priming experiments use picture production tasks, but simulating such a task is outside of the current model. Although word-referent information can be built into the model (see Johns & Jones, 2015 who simulated a number of visual world sentence processing effects with a distributional instance model), the point of this article is to determine how much complexity in human behavior can be accounted for with a simple instance modeling framework.

Instead of simulating results using the procedures of picture productions like in Bock (1986), instead an alternative, and simpler, procedure will be used in the following simulations.

Structural priming is found across a number of different experimental setups. For example, Potter and Lombardi (1998) demonstrated that similar results are found in sentence recall using rapid serial visual presentation (RSVP) structural priming. In an RSVP task, words are presented sequentially for a brief period of time, typically 100ms per word. After a brief intervening task, subjects are asked to recall the sentence that they saw. Similar to the results of Bock (1986), previously processed sentences impact the syntactic structure of the recalled sentence (Potter & Lombardi, 1998; Chang, Bock, & Goldberg, 2003).

To approximate this setup, the following will be done. The order representation of a prime sentence will be constructed with equation (2). This vector will serve as the order cue. A context vector will be constructed with equation (1), but with function words being removed. This will be the context cue. The function words were removed from the context vector to allow for a single cue to be used to differentiate different syntactic constructions. Both the context and order cue will be used to retrieve an echo vector, with instance activation calculated with equation (8). The echo will be used to determine the preferred syntactic construction, based on the cues and the information stored in memory.

Instead of using the production mechanism from the above simulations, here a 2-alternative forced choice (2AFC) test will be given to the model. Specifically, the order vectors of two possible syntactic constructions were constructed (e.g., “The brick struck the car’s windshield” versus “The car’s windshield was struck by a brick”). Then the similarity between the two order vectors and the retrieved echo was computed. Whichever order vector had the highest similarity to the echo was chosen. The previously used production method cannot be used in the following simulations because the length of some of the test sentences were too long for it to be successfully applied, due to the number of possible alternative constructions of longer sentences (e.g., for a sentence of 12 words there are 479,001,600 possible orderings of those words; see the General Discussion section for a more detailed examination of this problem). Given this combinatorial problem, it is not the case that the selected sentence type from the 2AFC task is the one most preferred overall by the model, but instead the selected sentence will be the one it prefers between the alternatives within a particular empirical manipulation.

In order to demonstrate how syntactic priming is being assessed in this framework, Figure 8 contains an outline of the simulation process. As described, there are two cues used: 1) an order cue which contains the order representation of a prime sentence and 2) a context cue which contains the content words of a sentence to be produced. These cues are jointly used to retrieve an order echo, which in turn contains latent information about how the content words should be ordered. This echo is used to differentiate between two different syntactic constructions of the same content words. If the unrelated prime sentence contained in the order cue (in this case, “*the corrupt inspector offered a deal to the bar owner*”) exerts an influence on the retrieval process then the order echo should be biased towards the alternative using the same syntactic construction as the prime (in this case, “*the lifeguard tossed a rope to the struggling child*”). If no syntactic priming is occurring in the model, then there should be no bias towards either construction.

In the following simulations, it will be assumed that the cue or priming information is kept in a different memory store than the instance memory store, which will be akin to a short-term or working memory store. The information contained in this store will bias the production mechanism, based on the similarity of the to-be-produced sentence and the previous sentences processed and stored in the short-term memory store. Clearly this is a simplification, as there is no mechanism for consolidation of currently experienced sentences into the instance memory store, for example. However, the current setup allows for an existence proof that an instance model of language can account for basic findings in syntactic priming, while future iterations of the modeling framework will need to modify the model to account for the flow of lexical information in both short- and long-term memory.

Previous simulations were trained on corpora with specific sentence lengths. Due to the following simulations using sentences sets with different number of words in them, this was not possible for the forthcoming simulations. Thus, the models will be trained on sentences of different sizes. Specifically, a corpus derived from fiction books (Johns, Jones, & Mewhort, 2019) will be used, with sentences of 6 to 13 words. Each sentence length will have 500,000 sentences included in the corpus, for a total corpus size of 4,000,000 sentences. In each individual simulation contained below, unless otherwise specified, each model will have 100,000 sentences contained in memory. Each of the 100,000 sentences were randomly sampled (meaning that in some samples a model could have longer sentences encoded and some could have shorter sentences). In order to overcome any issues to the randomness of the sentences contained in a model's memory, levels of priming were recorded across 50 resamples of the model's environmental vector and randomized sentence sets for each of the below simulations in order to get average performance. Vector size was reduced to N=1,024 to limit the computational costs of the model, unless otherwise specified.

Bock (1986). The first result simulated will be the classic results of Bock (1986). In this study, two types of sentence constructions were used: 1) either active transitive sentences (e.g., "*One of the fans punched the referee*") or passive transitive sentences (e.g. "*The referee was punched by one of the fans*"), or 2) prepositional dative sentences (e.g., "*The secretary is baking a cake for her boss*") or double object dative sentences (e.g., "*The secretary is baking her boss a cake*"). It was found that when people were primed by a sentence of a certain construction (e.g., "*The referee was punched by one of the fans*"), subjects were more likely to use that construction for a semantically unrelated sentence (e.g., "*The car's windshield was struck by a brick*") over an alternative construction (e.g. "*A brick struck the car's windshield*").

To simulate this effect, the sentences were taken directly from Bock (1986). For the dative sentences, there were 12 sentence pairs (one using a prepositional construction, the other using a double-object construction). For the transitive sentences, there were 24 sentence pairs (one using an active construction, the other a passive construction). Each of the dative sentences could act as a prime to another dative pair, and likewise for the transitive sentence pairs.

To simulate syntactic priming, the process displayed in Figure 8 was used. In particular, two sentence pairs were selected. One sentence pair will act as a prime, and one sentence pair will act as a target. The model is then primed with one sentence construction (e.g., a prepositional dative) from the sentence pair selected as a prime. For the target sentence, all function words were removed to be used as a context cue (so there is no bias towards either possible construction contained in the context cue). An order echo is retrieved, then the model is given a 2AFC task between the order representation for one syntactic construction (e.g., a propositional dative) and the representation of a different syntactic construction (e.g., a double object dative). The probability of selecting the primed sentence construction was then recorded. The process was then repeated using the other prime sentence (in this case, the double object dative) as a prime, and again the probability of selecting the primed sentence was recorded. To calculate overall level of priming seen, these probabilities were assessed by priming each sentence pair with all other sentence pairs (i.e., priming levels for each sentence pair was assessed for the 11 other sentence pairs for the dative sentences and 23 other sentence pairs for the transitive sentences), across 50 resamples.

To ensure that any changes in priming performance in the model was due to the prime construction, an unrelated priming condition was also used, where 36 sentences were taken from Bock and Griffin (2000) neutral prime sentence set.

The results of the simulation are contained in Figure 9. The left panel displays the probability of a prepositional dative construction being preferred by the model when primed with an unrelated sentence, an unrelated prepositional dative sentence, or an unrelated double-object dative sentence, while the right panel displays the same data for the active and passive transitive sentences. Given that the task is a 2AFC task, the probability of selecting the double-object construction is the compliment of the probabilities in the figure, and the same holds for passive sentences in the right panel. This figure shows that even though the prime and target sentences are semantically unrelated to each other, when cued with an order vector of a certain construction type, the retrieved echo is typically more similar to the primed construction. Additionally, there is a stronger effect for dative sentences compared to transitive sentences, which is also reflected in the data from Bock (1986). It is important to point out that the model that has no knowledge of syntax whatsoever. Instead the latent retrieval across thousands of instances of sentences produces structure that is coherent with what looks like higher-order syntax.

One question resulting from this simulation is what effect the number of sentences stored in memory has on the levels of priming seen in Figure 9. To test this, the level of priming seen from memory sizes 1,000 to 25,000 was tested. Levels of priming was assessed for both the dative and transitive sentence sets from Bock (1986). The priming level was assessed by taking the difference between the probability of selecting a propositional dative when primed with an unrelated propositional dative versus an unrelated double-object dative across different levels of memory storage, and vice versa for the double-object datives. An equivalent analysis was done for active over passive sentences for the transitive sentence set. The results of the simulation are contained in Figure 10. This figure shows that even at a low number of sentences (1,000 sentences) there is still priming, which suggests that there are structural similarities in the sentences of different grammatical categories that allow for syntactic priming to occur. However, there is also a substantial increase in the level of priming that occurs across memory size.

Bock and Griffin (2000). As stated previously, an important empirical result in structural priming is the finding that the effect is persistent across multiple intervening sentences, as shown conclusively by Bock and Griffin (2000). To simulate this effect, the dative sentences from Bock (1986) were used. The neutral prime sentences from Bock and Griffin (2000) were used as intervening sentences. The order cue will no longer just be the representation from the prime word, but will also contain the order vectors of up to 5 other sentences, simulating the effect of intervening sentences on the retrieval process. All other aspects of the simulation were kept identical to the previous simulation.

The assumption underlying this simulation is that short-term memory utilizes a composite vector representation, where all items are added into a single memory store. This assumption is made because of the fact that the representational mechanisms of the IPM are based off the architecture of the TODAM model of episodic memory (Murdock, 1982), which utilizes a composite representation of memory, where all items are added into a single vector. In TODAM, forgetting is not assumed to be due to decay of information within memory, but instead due to interference from other items stored (see Mewhort, et al., 2018 for a recent discussion of this issue). From this perspective, the effect of number of intervening sentences on priming is similar to retroactive interference in episodic memory performance (e.g., Baddeley & Dale, 1966), which is the impact of newly presented information on the remembering of previously presented information. By comparing levels of priming when there are no other sentences in memory to priming levels as other sentences are added into the memory, it signals the impact of retroactive interference on syntactic priming in the model. If the addition of other sentence forms into the short-term memory store eliminates priming it would signal that the model is not resilient to the presence of other syntactic forms in memory, inconsistent with empirical data. However, if the model is not overly impacted by the addition of other sentence forms into memory, it would demonstrate that the model is retaining information in memory about a prime, even though other information has been added into it, one of the benefits of using distributed representations (Kelly, Mewhort, & West, 2017).

The results are displayed in Figure 11, which shows that the amount of priming does decrease as an effect of the number of intervening items, but it is far from eliminated. This demonstrates that although the model is influenced by intervening items in memory, there is a residual effect of the prime sentence on retrieval. This result signals that the addition of other sentence representations into the short-term memory store does degrade the representation of the prime in memory somewhat, but it is not eliminated, and the model still shows considerable amounts of priming even with the addition of five other sentences into memory.

One related aspect of the model that has not been tested so far is the impact of vector size on model performance. This seems a particularly important manipulation for the current test of model performance as the storage capacity of a vector could impact the resolution of the prime sentence in memory. Given that there is no in principle way of determining the correct vector size for the model (indeed, larger vector sizes do not always lead to better model performance; Landauer & Dumais, 1997; Jones & Mewhort, 2007), the amount of priming across different vector sizes was computed. Since circular convolution uses fast Fourier transformations in its computations (Plate, 1995), vector size has to be a power of two. All vector sizes between 128 and 4,096 were tested. To calculate priming size, the average level of priming was calculated by computing the difference in probability in the model selecting the prepositional construction when primed with a prepositional sentence versus a double-object sentence, and vice versa. The resulting calculation signals the overall difference in levels of priming for the two primes. Figure 12 plots the results of this simulation when there are five intervening sentences in memory, and shows that the impact of the prime on construction selection is related to the vector size that is selected. Larger vector size allows for more retention of probe structure, leading to larger amounts of priming even with a substantial number of other sentences in memory.

However, one alternative explanation of Figure 12 is that larger vector sizes cause more priming overall. That is, it is not the greater level of prime retention that is causing the differences seen in Figure 12, but instead that the model has an overall increase in the level of priming as vector size is increased. To test this possibility, priming levels from vector sizes of 1,024 and 4,096 were contrasted with 0 to 5 intervening sentences placed into memory. If there is an overall greater level of priming for the larger vector size at each level, then it would signal that the larger vector size is causing an overall increase in priming. The results of the simulation are contained in Figure 13, and shows that the priming between the two vector sizes are equivalent when there are no other sentences in memory, but that the larger vector size maintains priming levels across the number of intervening sentences better than the smaller vector size. This result demonstrates that it is the greater retention of prime information at larger vector sizes that is causing the greater level of priming. The ability of the model to be resistant to the impact of intervening items is likely due to the nature of the distributed representations that the model employs (see Franklin & Mewhort, 2015; Jones & Mewhort, 2007; Kelly, Mewhort, & West, 2017; Mewhort, Shabahang, & Franklin, 2017).

Lexical boost. A consistent finding in the structural priming literature is the lexical boost phenomenon, where the presence of lexical overlap (typically through the prime and target sentence using overlapping verbs) leads to larger amounts of priming (Branigan, Pickering, & Cleland, 2000; Pickering & Branigan, 1998). This finding suggests that people are more likely to use a previously processed construction if there is word overlap between that construction and the current words being produced. This finding is a natural data type for the IPM to explain since the inclusion of overlapping words in the cue should preferentially active instances using that verb in that structure.

To simulate this finding, sentences were taken from Rowland, Chang, Ambridge, Pine, and Lieven (2012). Rowland, et al. assembled sentence sets for six verbs. The sentence set used in the following simulation was the sentence set not using proper nouns, which consisted of twelve sentences that could have either a prepositional dative or a double-object dative construction. Thus, for each verb there were four sentences – two prepositional datives and two double-object datives. Each sentence served as a target, and were equally primed by both constructions. Prime sentences were either a randomly selected sentence using a different verb (mismatching verb condition), or the unrelated sentence with the same verb (matching verb condition). For example, for the target sentence "*The boy passed the girl a fish*" could be primed by the sentence "*The king threw the queen a rabbit*" in the mismatching condition, and the sentence "*The king passed a baby to the queen*" in the matching condition. Since the Rowland, et al. study is a developmental one, the sentences are relatively simpler than the previous sentence sets used. Thus, to keep model performance from being at ceiling, 10 intervening sentences (neutral prime sentences from Bock & Griffin, 2000) were added into the order cue. All other simulation details were kept equivalent. Priming levels were assessed by taking the difference in the level of priming when there was a matching versus a mismatching verb. That is, the difference in the level of priming was assessed when the prime was a prepositional dative with a matching verb versus a prepositional dative with a mismatching verb. The equivalent measure was assessed for the double object datives, and the level of priming observed for the two sentence types were collapsed together to show an overall level of priming when a mismatched and matched verb is used.

Results of the simulation are contained in Figure 14. This figure shows that there are still high levels of priming when using a mismatching verb, consistent with previous simulations showing structural priming even without verb overlap in sentences. However, the inclusion of an overlapping verb significantly increases the amount of priming. This demonstrates that the inclusion of a verb in both the context and order cue provides complementary environmental information about the correct construction to use, leading to an incremental increase of the similarity between the retrieved echo and the representation of that construction.

Chang, Bock, and Goldberg (2003). So far, it has been demonstrated that the IPM can account for structural priming at the sentence level, where cueing memory with the order vector of an unrelated structure causes that structure to be retrieved from memory more readily. However, structural priming has also been demonstrated within a single sentence construction as well. Specifically, Chang, Bock, and Goldberg (2003) conducted a sentence recall experiment using spray-load alternation sentences (Anderson, 1971). In a spray-load sentence (e.g., "*The farmer heaped straw onto the wagon*"), the theme is the object that moves (*straw*) and the location is where that object is moved (*wagon*). With the spray-load alternation, sentences can either by theme-location ordered (e.g., "*The farmer heaped straw onto the wagon*") or location-theme ordered (e.g., "*The farmer heaped the wagon with straw*"). Chang, et al. (2003) used an RSVP structural priming task, where subjects read briefly presented spray-load sentences that were theme-location or location-theme ordered, then were given a distraction task. Subjects were then asked to recall the studied sentence. When primed with an unrelated sentence with a different ordering, subjects would occasionally recall the studied sentence with the prime ordering, rather than the actual ordering.

To simulate this finding, the spray-load sentences from Chang, et al. (2003) were used. The sentence set included 32 sentences that could have either a theme-location or location-theme ordering. To simulate this finding, the order cue included a prime sentence (a randomly selected sentence in either a theme-location or location-theme ordering), the studied sentence, and random Gaussian noise (with the same properties as the environmental vectors for words, but with a mean of 2.0). The noise is assumed to take the place of the distractor task in the experimental setup. The context cue is the studied sentence, minus function words, equivalent to previous simulations. The similarity between the retrieved echo and the theme-location and location-theme ordering of the studied sentence was taken, and the ordering with the highest similarity was selected. All other simulation details were kept the same as previous simulations.

The results of the simulation are contained in Figure 15. This figure shows that when the model is primed with an unrelated sentence that had a different theme and location ordering than the studied sentence, the model preferred that ordering to the studied ordering on a small percentage of trials, consistent with the data from Chang, et al. (2003). This simulation demonstrates that the model can account for both priming results across syntactic constructions, and also the ordering of information within sentences, consistent with empirical findings.

Discussion. The simulations contained in this section demonstrates conclusively that an instance model provides a promising modeling framework for accounting for structural priming. Although the model is not as developed as the model of Chang, et al. (2006), it also does not make any assumptions about abstracted linguistic information. Instead, the model relies upon the latent structure contained in a retrieved vector across thousands of individual instances of natural language. Additionally, it calls into question the notion that structural priming is a form of implicit learning, and that it is not memory based. The ability of the IPM to account for structural priming by simply using a short-term store of recently used language demonstrates that a memory-based explanation of this phenomenon does work.

Additionally, the success of the model naturally fits into usage-based and adaptive approaches to language processing. Specifically, in order to account for structural priming, the IPM proposes that in order to optimize communicative effectiveness, one should communicate like others are communicating in the current context. The model's context cue is a representation of what needs to be communicated, while the order cue provides information about how it should be structured (based on how others are communicating in context). Instead of a build-up of abstracted syntactic information in memory to account for priming, the IPM proposes that priming is an example of the adaptivity of language, where people use the structure of the current context to retrieve language that is structured in a similar way to how language in our current communicative environment is structured.

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 aspects of language, many have proposed that a formal grammar is necessary. A formal grammar is a top-down mechanism that seeks to understand language processing in light of abstract categories. Our approach, illustrated with the Instance Production Model (IPM), suggests that ordering need not depend on application of grammatical rules but rather depends on the structure of past utterances that one previously experienced. Under this approach, past experience informs and constrains future behavior. The IPM was designed to exploit the productivity and regularity of natural language, in order to illustrate the power of experience in producing grammatical utterances.

It is not an analysis or encoding of a single utterance that provides the knowledge needed to produce syntactic utterances; instead, it is the overlap in the usage of language. Even though no two utterances may be identical, the structure of a language emerges as a function of recorded instances in the act of cue-driven parallel retrieval from memory. The regular, but not identical, structure of studied utterances affords grammar-like behavior, albeit without an actual grammar.

The IPM is a simple model that encodes pure location and linear n-gram information to encode an instance of a sentence. A classic instance memory retrieval operation, grounded in principles of MINERVA 2 (Hintzman, 1986, 1988), 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 the one that is 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 previous efforts to build an instance-based model of language learning and comprehension using the same mechanisms and ideas, especially from the usage-based perspective (see Abbot-Smith & Tomasello, 2006; 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 instance-based model of memory can serve as a valuable competitor in the discipline's pursuit of a theory of integrated models of language comprehension and production.

The model successfully constructed the correct ordering of simple sentences of lengths 3 to 7 to a high degree, with a small linear drop in performance as sentence length increases. Additionally, the model was able to fill-in missing function words (and also determine the correct ordering of those words, a task which vastly increases the search size of the problem), and generalize to a different language.

However, the really interesting part of the model's behavior is the performance of the IPM as a function of the number of instances it has studied. Performance rapidly improves with the first 50,000 sentences studied, but then sees only small improvements as additional sentences are stored. This provides a look into what the regular nature of language provides to the act of productivity: even with a small range of linguistic experience, the syntactic regularities in language become apparent. Language is far from random and its redundancy allows a simple model that capitalizes on redundancy to construct syntactically correct sentences without any higher-level processing. As more instances 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.

Additionally, the ability of the model to account for structural priming data illustrates that the cued retrieval technique that the model utilizes is able to retrieve syntactic structure. It also demonstrates that the model is capable of accounting for various empirical examinations into language production. Importantly, it is able to do so without any assumptions about stimuli – sentences were taken directly from the studies without any need to manipulate them, one of the advantages of using corpus-based models of cognition (Johns, Jamieson, & Jones, 2020). The structural priming simulations also further demonstrated the connection between instance models and theoretical perspectives within the language sciences, particularly usage-based (Tomasello, 2001) and adaptive (Beckner, et al., 2009) approaches, as the underlying conceptualizing of syntactic priming that the model proposes is based on increasing communicate effectiveness through adaptation to local linguistic context.

Nevertheless, the instance-based approach to language has challenges. For example, the model might be charged with operating at the wrong level of analysis – phrases may be the right unit of language traces 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 instance construction process. This would also allow the model to operate with a smaller number of words; advantageous in terms of computational efficiency. It would also allow for the model to be tested on longer sentence lengths, as using the production mechanism described here quickly runs into a combinatorial problem. By constructing instances at the phrase level, the same production mechanism could be used while reducing the computational requirements.

A related issue with the model concerns its encoding scheme. In the simulations presented, memory was populated with sentences of the same length as the test sentences. Model performance is influenced by this because it reduces the impact of word location information, which Figure 1 demonstrates is an important information source for the model. More research is required to determine the best mechanism to encode location in a relative fashion, where sentences of different lengths are included in the same retrieval process. Moving to encoding at the phrase level could alleviate these issues, as phrases are typically much shorter than full sentences. However, breaking sentences into phrases would require integrating grammatical knowledge into instance construction. The inclusion of grammatical class information, such as the ability to form a hierarchical representation of a sentence by parsing noun and verb phrases would undoubtedly improve the performance of the model, and allow it to process more complex sentence types. However, it would also significantly increase the amount of linguistic knowledge that is being built into the model, which was intentionally avoided in the simulations reported in this article, in order to demonstrate the base power and usefulness of an instance approach to language production. Thus, future modeling efforts will need to balance model performance with model complexity.

However, these problems arise because of the approach's simplicity, 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. That is, the IPM is not a complete explanation of how humans are capable of producing language, as there are many phenomena that are beyond the model's current abilities. Instead, the results reported here are an existence proof of sorts, demonstrating that an instance model with very little complexity built into it can explain complex linguistic phenomena. Future work will need to determine what else needs to be built into the framework to allow it to be a more complete model.

In closing, we wish to comment on how the work speaks to two larger issues: the evolution of language and the relationship between our understanding of language and human memory. Christiansen and Chater (2008) present a framework of language evolution in which language evolved by overtaking or using other cognitive systems, which includes memory processing. Our results lend credence to this theoretical approach. The IPM uses standard memory processing techniques (Hintzman, 1986, 1988) to generate complex linguistic behaviors. Use of techniques that have been developed in the memory field can be readily applied to solve problems within language processing (Jamieson & Mewhort, 2009, 2010, 2011; Jamieson, et al., 2018; Johns & Jones, 2015). The opposite is also true, where language processing models can be integrated into memory models to provide content for a memory model to utilize (Johns & Jones, 2010; Johns, Jones, & Mewhort, 2012; Mewhort, Shabahang, & Franklin, 2018). That is, memory and language processing are not separate systems, and theories in both fields can be enhanced by examining the types of formal mechanisms that allow for an increased integration of the two cognitive processes.

References

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

Anderson, S. R. (1971). On the role of deep structure in semantic interpretation. *Foundations of Language*, 7, 387-396.

Bannard, C., Lieven, E., & Tomasello, M. (2009). Modeling children's early grammatical knowledge. *Proceedings of the National Academy of Sciences, 106,* 17284-17289.

Beckner, C., Blythe, R., Bybee, J., Christiansen, M. H., Croft, W., ... & Schoenemann, T. (2009). Language is a complex adaptive system: Position paper. *Language Learning*, 59, 1-26.

Bernolet, S., Collina, S., & Hartsuiker, R. J. (2016). The persistence of syntactic priming revisited. *Journal of Memory and Language*, 91, 99-116.

Berwick, R. C., Pietroski, P., Yankama, B., & Chomsky, N. (2011). Poverty of the stimulus revisited. *Cognitive Science*, 35, 1207-1242.

Bock, J. K. (1986). Syntactic persistence in language production. *Cognitive Psychology*, 18, 355-387.

Bock, K., & Griffin, Z. M. (2000). The persistence of structural priming: Transient activation or implicit learning? *Journal of Experimental Psychology: General*, 129, 177.

Branigan, H. P., Pickering, M. J., & Cleland, A. A. (2000). Syntactic co-ordination in dialogue. *Cognition*, 75, B13-B25.

Brooks, L. R. (1978). Nonanalytic Concept Formation and Memory for Instances. In E. Rosch & B. Lloyd (Eds.), *Cognition and Categorization* (pp. 3–170). Hillside, NJ: Lawrence Elbaum Associates.

Brooks, L. R. and Vokey, J. R. 1991. Abstract analogies and abstracted grammars: Comments on Reber (1989) and Mathews et al. (1989). *Journal of Experimental Psychology: General*, 120, 316–320.

Brysbaert, M., Stevens, M., Mandera, P., & Keuleers, E. (2016). How many words do we know? Practical estimates of vocabulary size dependent on word definition, the degree of language input and the participant's age. *Frontiers in Psychology*, 7, 1116.

Chang, F., Bock, K., & Goldberg, A. E. (2003). Can thematic roles leave traces of their places? *Cognition*, 90, 29-49.

Chang, F., Dell, G. S., & Bock, K. (2006). Becoming syntactic. *Psychological Review*, 113, 234-272.

Chang, F., Lieven, E., & Tomasello, M. (2008). Automatic evaluation of syntactic learners in typologically-different languages. *Cognitive Systems Research*, 9, 198-213.

Chomsky, N. (1957). *Syntactic structures*. The Hague: Moutin.

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

Chomsky, N. (1991). Linguistics and cognitive science: Problems and mysteries. In A. Kasher(Ed.), *The Chomskyan turn*. Cambridge, MA: Blackwell.

Christiansen, M. H., & Chater, N. (2008). Language as shaped by the brain. *Behavioral and Brain Sciences, 31*, 489-509.

Christiansen, M. H., & Chater, N. (2016). The Now-or-Never bottleneck: A fundamental constraint on language. *Behavioral and Brain Sciences*, *39*, e62.

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

Chubala, C. M., Johns, B. T., Jamieson, R. K., & Mewhort, D. J. K. (2016). Applying an instance model to an implicit rule-learning task: Implicit learning of semantic structure. *The Quarterly Journal of Experimental Psychology*, *69*, 1049-1055.

Cleeremans, A., & McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General*, 120, 235.

Crump, M. J. C., Jamieson, R. K., Johns, B. T., & Jones, M. N. (2020). Controlling the retrieval of general versus specific semantic knowledge in the instance theory of semantic memory. *Proceedings of the 42nd Annual Cognitive Science Society*.

Dalton, P. (1993). The role of stimulus familiarity in context-dependent recognition. *Memory & Cognition*, 21, 223-234.

Dell, G. S., & Ferreira, V. S. (2016). Thirty years of structural priming: An introduction to the special issue. *Journal of Memory and Language*, 91, 1-4.

Dienes, Z., Altmann, G. T., & Gao, S. J. (1999). Mapping across domains without feedback: A neural network model of transfer of implicit knowledge. *Cognitive Science*, 23, 53-82.

Dosher, B. A., & Rosedale, G. S. (1997). Configural processing in memory retrieval: Multiple cues and ensemble representations. *Cognitive Psychology*, 33, 209-265.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14, 179-211.

Franklin, D. R., & Mewhort, D. J. K. (2015). Memory as a hologram: An analysis of learning and recall. *Canadian Journal of Experimental Psychology*, 69, 115-135.

Gold, E. M. (1967). Language identification in the limit. *Information and control, 10*, 447-474.

Goldinger, S. D. (1998). Echoes of echoes? An episodic trace theory of lexical access. *Psychological Review*, 105, 251-279.

Hare, M., Jones, M., Thomson, C., Kelly, S., & McRae, K. (2009). Activating event knowledge. *Cognition*, *111*, 151-167.

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

Hintzman, D. L. (1988). Judgments of frequency and recognition memory in a multiple-trace memory model. *Psychological Review, 95*, 528-551.

Jamieson, R. K., & Mewhort, D. J. K. (2005). The influence of grammatical, local, and organizational redundancy on implicit learning: An analysis using information theory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *31*, 9–23.

Jamieson, R. K., & Mewhort, D. J. K. (2009). Applying an instance 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 instance 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.

Jamieson, R. K., Avery, J. E., Johns, B. T., & Jones, M. N. (2018). An instance theory of semantic memory. *Computational Brain and Behavior*, 2, 119-136.

Johns, B. T. & Jones, M. N. (2010). Evaluating the random representation assumption of lexical semantics in cognitive models. *Psychonomic Bulletin and Review*, *17,* 662-672.

Johns, B. T., Jones, M. N., & Mewhort, D. J. K. (2012). A synchronization account of false recognition. *Cognitive Psychology, 65*, 486-518.

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.

Johns, B. T., Jones, M. N., & Mewhort, D. J. K. (2016). Experience as a free parameter in the cognitive modeling of language. *Proceedings of the 37th Annual Conference of the Cognitive Science Society*.

Johns, B. T., Mewhort, D. J. K., & Jones, M. N. (2017). Small worlds and big data: Examining the simplification assumption in cognitive modeling. In M. Jones (Ed.), *Big Data in Cognitive Science*. Taylor & Francis: Psychology Press.

Johns, B. T., Taler, V., Pisoni, D. B., Farlow, M. R., Hake, A. M., Kareken, D. A., Unverzagt, F. W., & Jones, M. N. (2018). Cognitive modeling as an interface between brain and behavior: Measuring the semantic decline in mild cognitive impairment. *Canadian Journal of Experimental Psychology*, 72, 117-126.

Johns, B. T., Jamieson, R. K., & Jones, M. N. (in press). The continued importance of theory: Lessons from big data approaches to cognition. In Woo, S. E., Proctor, R., and Tay, L. (Eds.) *Big Data Methods for Psychological Research: New horizons and Challenges*, APA Books.

Johns, B. T., Jones, M. N., & Mewhort, D. J. K. (2019). Using experiential optimization to build lexical representations. *Psychonomic Bulletin & Review*, 26, 103-126.

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

Jones, M. N., Willits, J. A., & Dennis, S. (2015). Models of semantic memory. In J. R. Busemeyer, Z. Wang, J. T. Townsend, & A. Eidels (Eds.) *Oxford Handbook of Mathematical and Computational Psychology*. (pp. 232-254). New York: OUP.

Jones, M. N., Dye, M., & Johns, B. T. (2017). Context as an organizational principle of the lexicon. In B. Ross (Ed.), *The Psychology of Learning and Motivation*, 67:43.

Jones, M. N. (2017). *Big Data in Cognitive Science*. Psychology Press: Taylor & Francis.

Jones, M. N. (2018). When does abstraction occur in semantic memory: Insights from distributional models. *Language, Cognition and Neuroscience*.

Jones, M. N., Johns, B. T., & Recchia, G. (2012). The role of semantic distinctiveness in lexical organization. *Canadian Journal of Experimental Psychology*, 66, 115-124.

Kelly, M. A., Blostein, D., & Mewhort, D. J. K. (2013). Encoding structure in holographic reduced representations. *Canadian Journal of Experimental Psychology*, 67, 79-93.

Kelly, M. A., Reitter, D., & West, R. L. (2017, July). Degrees of separation in semantic and syntactic relationships. In *Proceedings of the 15th International Conference on Cognitive Modeling*.

Kelly, M. A., Mewhort, D. J., & West, R. L. (2017). The memory tesseract: Mathematical equivalence between composite and separate storage memory models. *Journal of Mathematical Psychology*, 77, 142-155.

Kinder, A. (2010). Is grammaticality inferred from global similarity? Comment on Jamieson & Mewhort (2009). *Quarterly Journal of Experimental Psychology*, 63, 1049–1056.

Kinder, A., & Lotz, A. (2009). Connectionist models of artificial grammar learning: what type of knowledge is acquired? *Psychological Research PRPF*, 73, 659-673.

Kruschke, J. K. (1992). ALCOVE: An instance-based connectionist model of category learning. *Psychological Review*, *99*, 22.

Kwantes, P. J. (2005). Using context to build semantics. *Psychonomic Bulletin & Review*, *12*, 703-710.

Kwantes, P. J., & Mewhort, D. J. K. (1999). Modeling lexical decision and word naming as a retrieval process. *Canadian Journal of Experimental Psychology*, 53, 306-315.

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.

Laurence, S., & Margolis, E. (2001). The poverty of the stimulus argument. *The British Journal for the Philosophy of Science*, *52*, 217-276.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*, 492-527.

Logan, G. D. (2002). An instance theory of attention and memory. *Psychological Review, 109*, 376-400.

Nosofsky, R. M. (1986). Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology: General*, *115*, 39-57.

Nosofsky, R. M. (1991). Tests of an instance model for relating perceptual classification and recognition memory. *Journal of Experimental Psychology: Human Perception and Performance*, *17*, 3-27.

Mahowald, K., James, A., Futrell, R., & Gibson, E. (2016). A meta-analysis of syntactic priming in language production. *Journal of Memory and Language*, 91, 5-27.

Mehl, M. R., Vazire, S., Ramírez-Esparza, N., Slatcher, R. B., & Pennebaker, J. W. (2007). Are women really more talkative than men? *Science*, 317, 82-82.

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.

Mewhort, D. J. K., Shabahang, K. D., & Franklin. D. R. J. (2018). Release from PI: An analysis and a model. *Psychonomic Bulletin & Review*, 25, 932-950.

Murdock, B. B. (1982). A theory for the storage and retrieval of item and associative information. *Psychological Review*, 89, 609-626.

Murdock, B. B. (1983). A distributed memory model for serial-order information. *Psychological Review*, 90, 316.

Osth, A. F., Shabahang, K. D., Mewhort, D. J., & Heathcote, A. (2020). Global semantic similarity effects in recognition memory: Insights from BEAGLE representations and the diffusion decision model. *Journal of Memory and Language*, 111, 104071.

Perfors, A., Tenenbaum, J., & Regier, T. (2006). Poverty of the stimulus? A rational approach. *Proceedings of the Cognitive Science Society*.

Pickering, M. J., & Branigan, H. P. (1998). The representation of verbs: Evidence from syntactic priming in language production. *Journal of Memory and Language*, 39, 633-651.

Pickering, M. J., & Ferreira, V. S. (2008). Structural priming: A critical review. *Psychological Bulletin*, 134, 427-459.

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

Potter, M. C., & Lombardi, L. (1998). Syntactic priming in immediate recall of sentences. *Journal of Memory and Language*, 38, 265-282.

Qui, M. & Johns, B. T. (2020). Semantic diversity in paired-associate learning: Further evidence for the information accumulation perspective of cognitive aging. *Psychonomic Bulletin & Review*, 27, 114-121.

Quine, W. V. O. (1960). *Word and object*. Cambridge, MA: MIT Press.

Ratcliff, R., & McKoon, G. (1994). Retrieving information from memory: Spreading-activation theories versus compound-cue theories. *Psychological Review*, 101, 177–184.

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.

Recchia, G. L., Sahlgren, M., Kanerva, P., & Jones, M. N. (2015). Encoding sequential information in vector space models of semantics: Comparing holographic reduced representation and random permutation. *Computational Intelligence & Neuroscience*.

Reitter, D., Keller, F., & Moore, J. D. (2011). A computational cognitive model of syntactic priming. *Cognitive Science*, 35, 587-637.

Robins, R. H. (2013). *A short history of linguistics*. Routledge: New York.

Rosch, E. H. (1973). Natural categories. *Cognitive Psychology*, *4*, 328-350.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., & Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology, 8*, 382-439.

Rowland, C. F., Chang, F., Ambridge, B., Pine, J. M., & Lieven, E. V. (2012). The development of abstract syntax: Evidence from structural priming and the lexical boost. *Cognition*, 125, 49-63.

Sahlgren, M. (2006). The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces (Doctoral dissertation, Institutionen för lingvistik).

Shepard, R. N. (1987). Towards a universal law of generalization for psychological science. *Science*, 237, 1317-1323.

Thiessen, E. D., & Pavlik Jr, P. I. (2013). iMinerva: A mathematical model of distributional statistical learning. *Cognitive Science*, 37, 310-343.

Tomasello, M. (2000). Do young children have adult syntactic competence? *Cognition*, *74*, 209-253.

Tomasello M. (2003). *Constructing a language: A usage-based theory of language acquisition*. Cambridge, MA: Harvard Univ. Press.

Table 1. Example of test sentences.

Length	Sentence
	She was crying.
3	It sounds better.
	I felt dreadful.
	She held her breath.
4	I'm out of here.
	They ruined his life.
	He took a deep breath.
5	Let me tell my story.
	My heart was beating fast.
	I'll meet you at the house.
6	I had no intention of offending.
	People stared at me in silence.
	He had never seen anything like it.
7	I'm not going to stand for it.
	I'd never seen such an ugly cat.

Table 2. Example of model output across training for different sentences of six words.

Memory Size	Model Output
1,000	Dog the good was a boy.
4,000	The boy dog was a good.
37,000	The boy was a good dog.
172,000	**The dog was a good boy.**
1,000	Smells like to me it trouble.
8,000	Smells like it to me trouble.
32,000	Like trouble smells it to me.
183,000	**It smells like trouble to me.**
1,000	He had no dying of fear.
9,000	He had dying of no fear.
44,000	**He had no fear of dying.**
1,000	Home bring we to her need.
15,000	Home need we to her bring.
88,000	Home we need to her bring.
155,000	We need bring to her home.
164,000	**We need to bring her home.**
1,000	The poked in you eyes him.
5,000	The him in you eyes poked.
28,000	**You poked him in the eyes.**
1,000	Suffering depression was from the man.
19,000	Suffering man was from the depression.
236,000	The man from depression was suffering.
336,000	**The man was suffering from depression.**
1,000	Offending I had no intention of.
83,000	I had no of offending intention.
101,000	**I had no intention of offending.**

Note. Correct sentence ordering in bold.

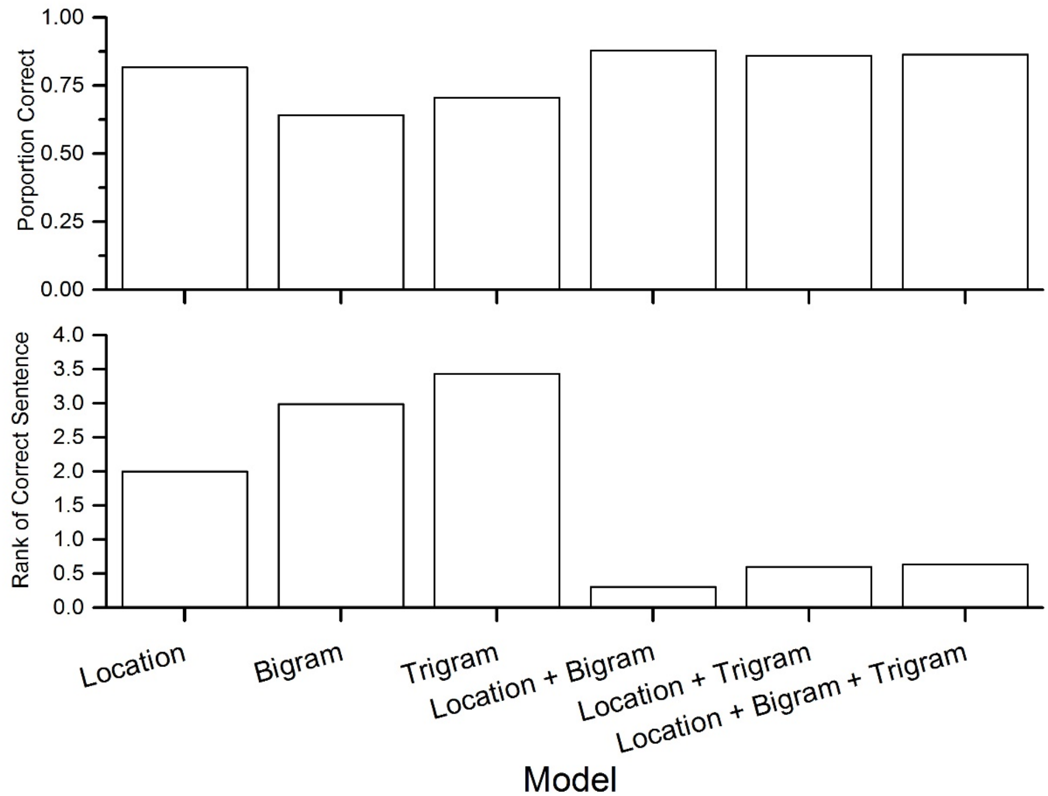

Figure 1. A comparison of model performance using different combinations of lexical information sources on two tests – proportion correct (top panel) and rank of the correct sentence (bottom panel). The resulting simulation shows that the most parsimonious model is locative information combined with bigram information.

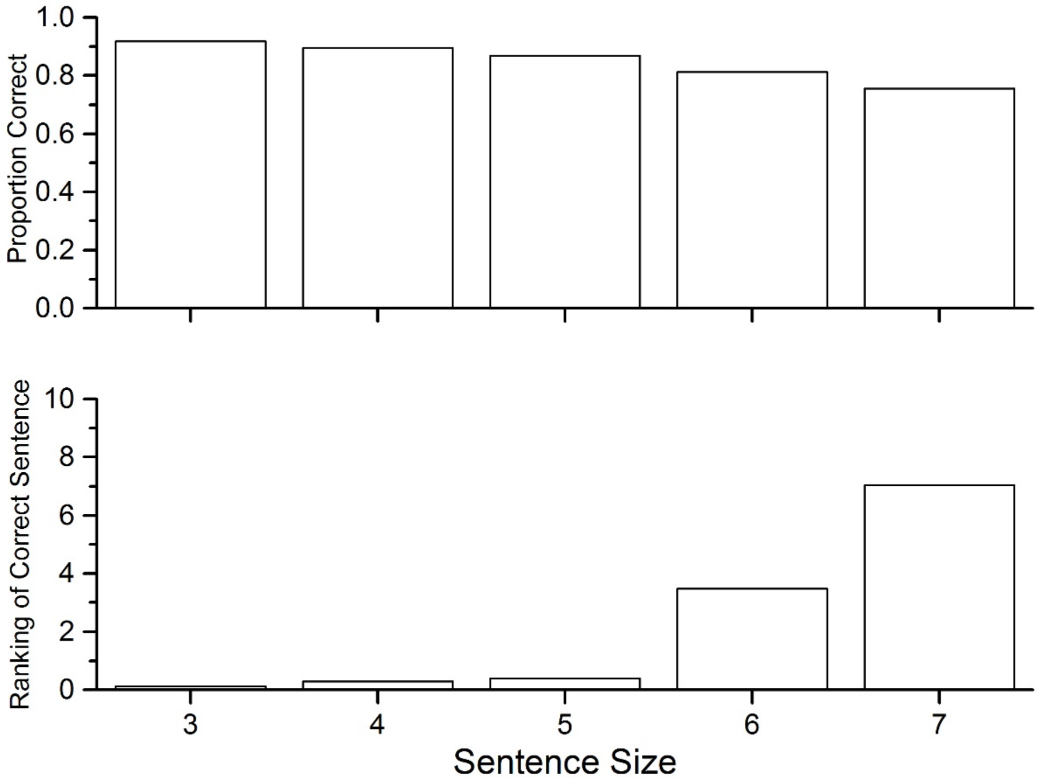

Figure 2. Results of the IPM on sentences of lengths three to seven, on the two tests of model performance.

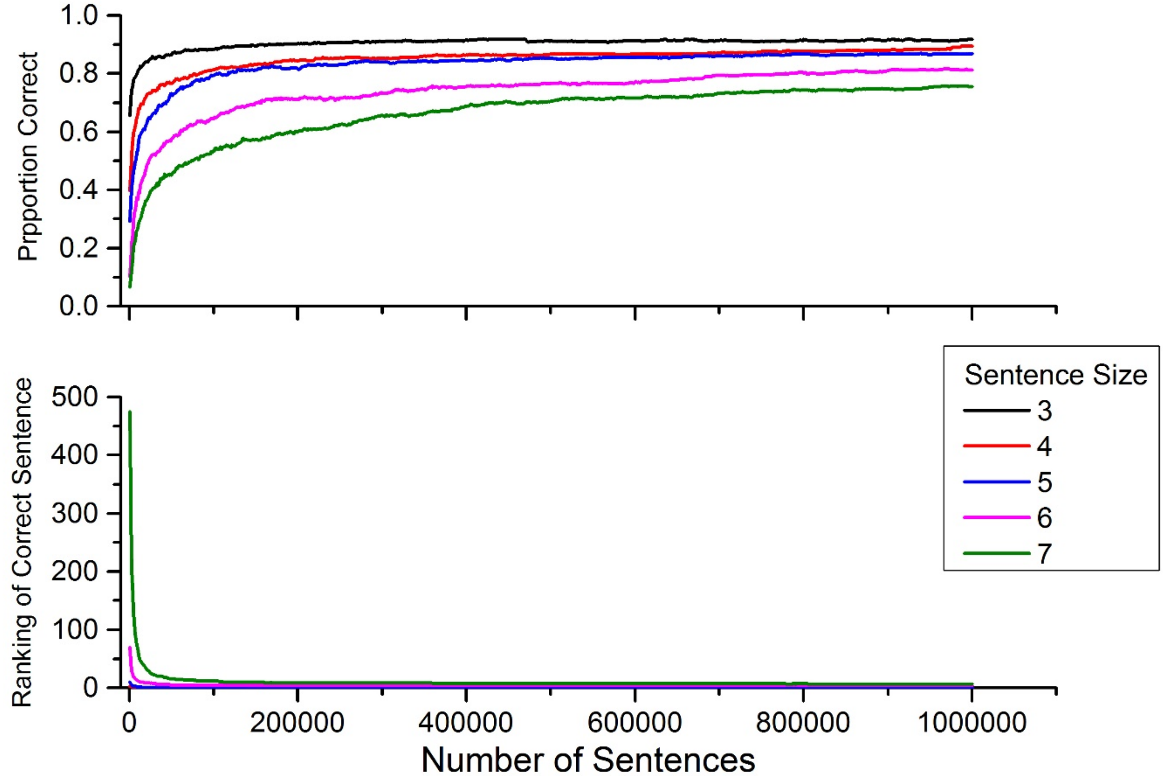

Figure 3. The time course of model performance. This simulation demonstrates that most of the change happens with very few lexical experiences (<100,000), with small improvements for subsequent instances.

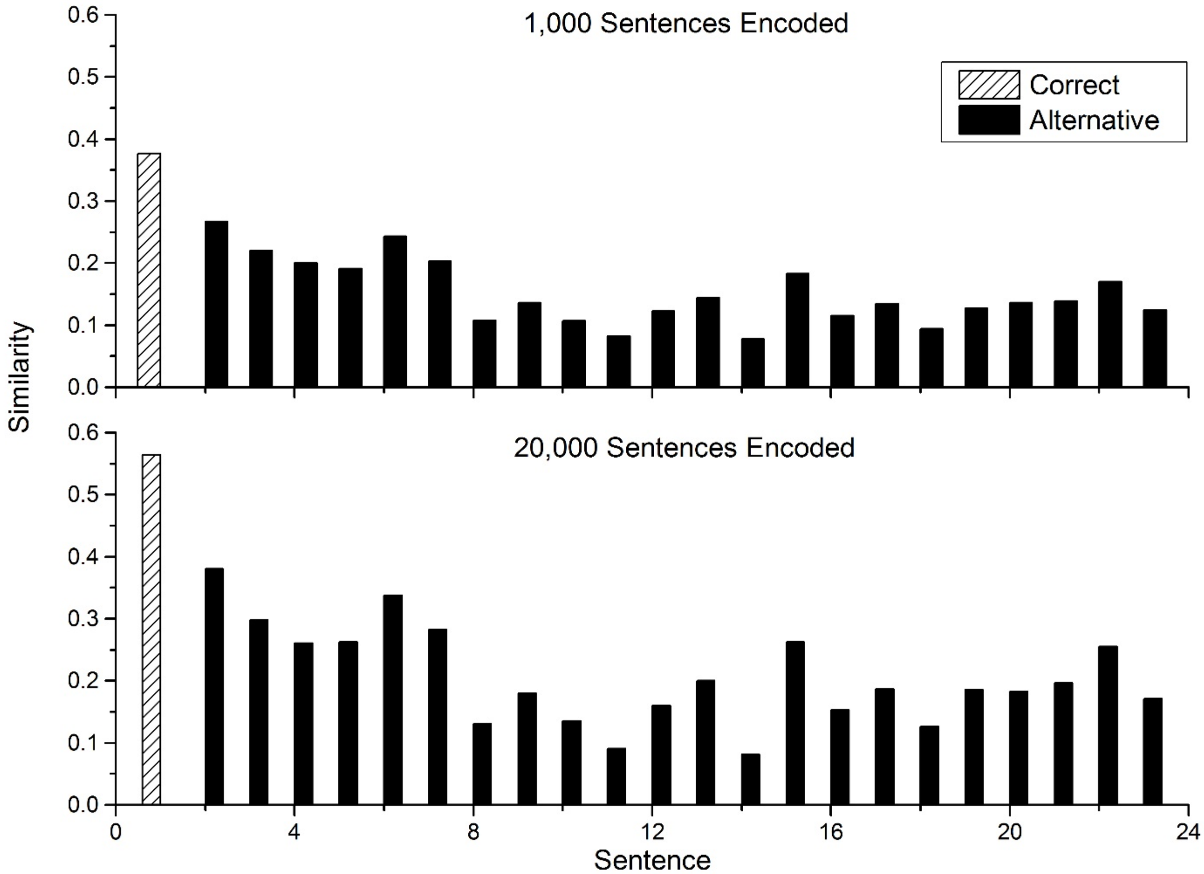

Figure 4. A comparison of the relative activation levels of the correct ordering of a sentence and the possible alternatives at 1,000 and 20,000 sentences recorded for four word sentences. The similarity displayed in the figure is between the retrieved echo from the sentences context vectors and the order representation for the correct ordering and the possible alternatives.

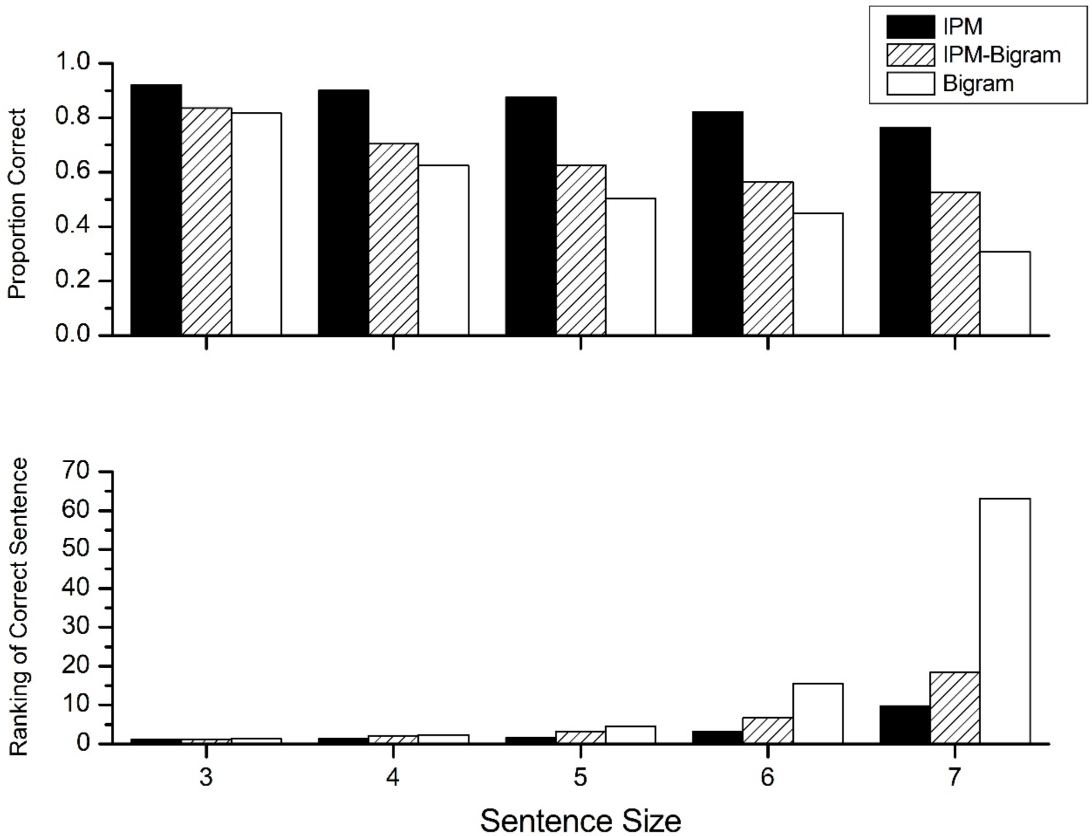

Figure 5. Comparison of the IPM and IPM with bigram-only information compared to a standard bigram probability model. This demonstrates that the multiple types of information encoded, and the retrieval mechanism, allow for the model to have an increased level of performance compared to a standard model type.

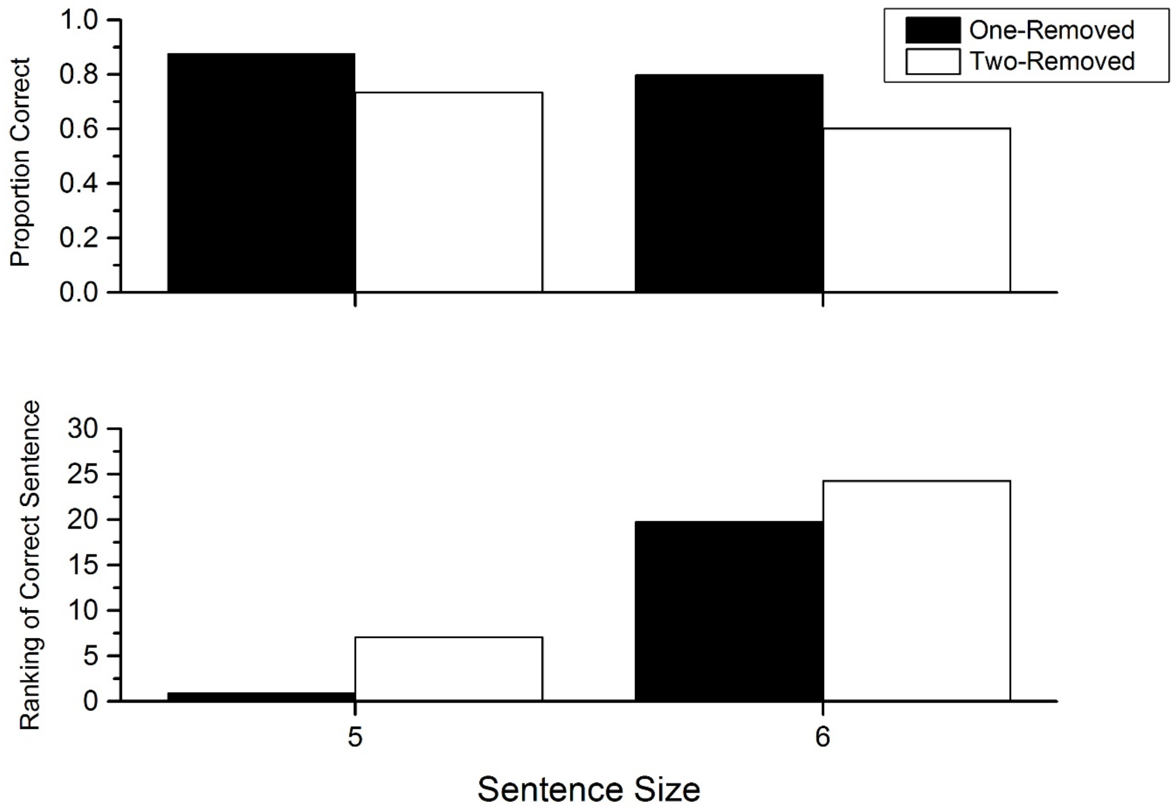

Figure 6. Results of the filling-in simulation, where the model had to generate both the missing function word/words and the resulting ordering of those words, greatly increasing the number of possible orderings.

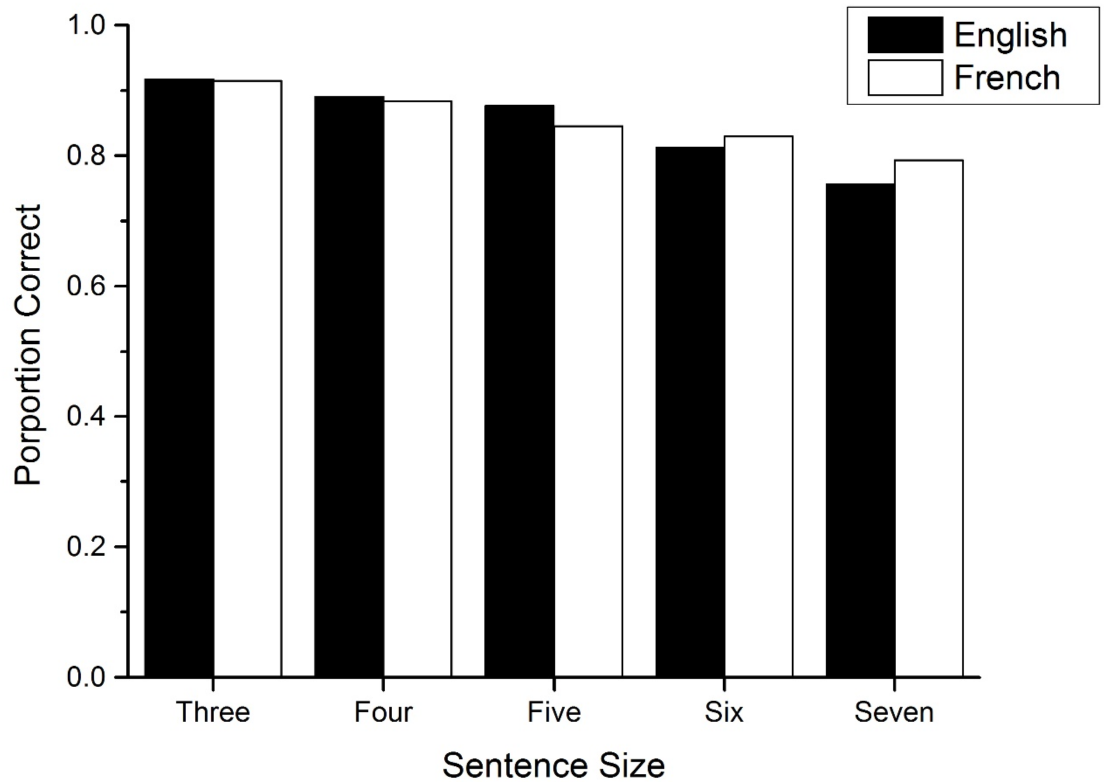

Figure 7. Model performance when trained and tested on an English corpus versus performance when trained and tested on a French corpus.

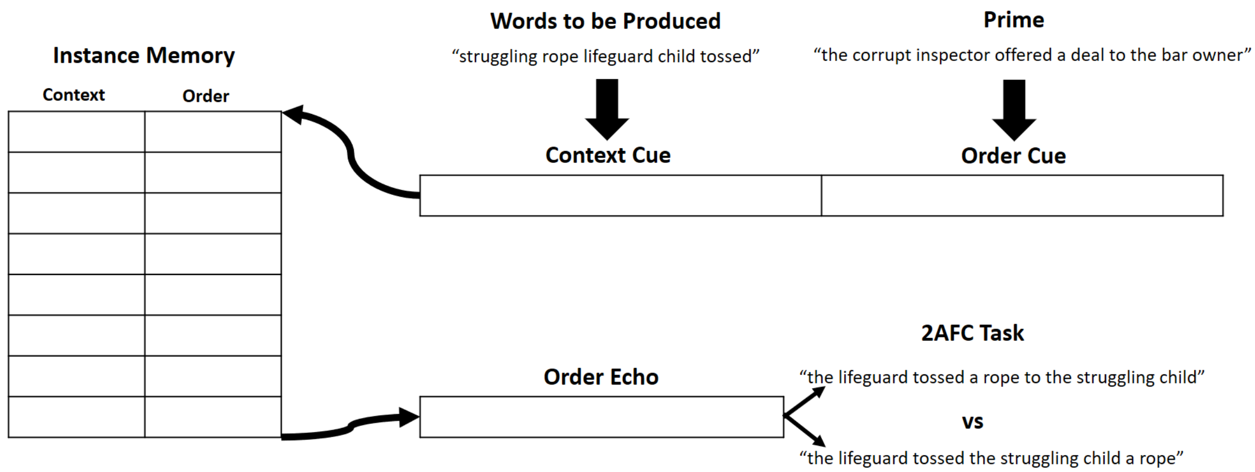

Figure 8. A pictorial demonstration of the syntactic priming task given to the IPM. The order representation of a prime sentence is used as an order cue, while the context cue is the summation of the content words needing to be produced minus any function words. The joint cues are used to retrieve an order echo, which is used in a 2AFC task where the model has to differentiate between two possible syntactic constructions.

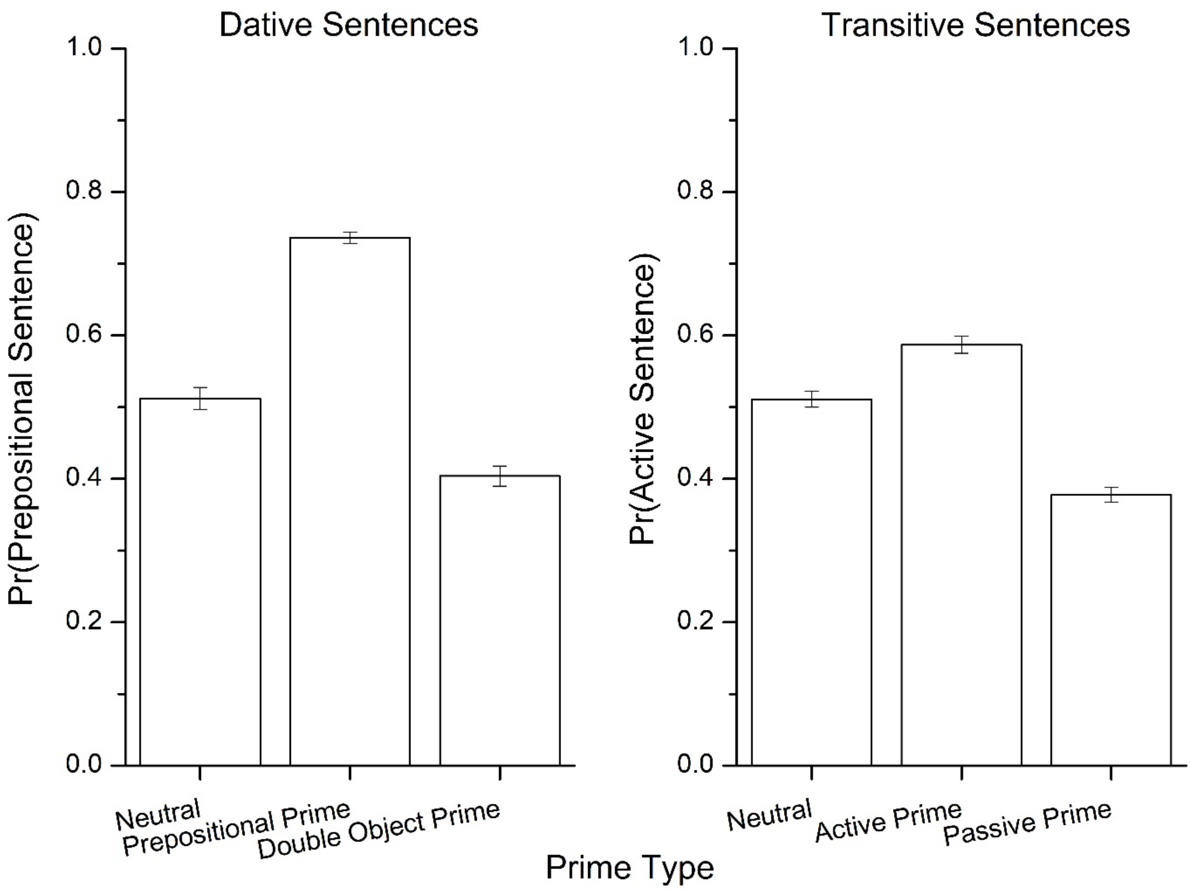

Figure 9. Simulation of Bock (1986). This figure displays the probability of selecting the prepositional construction (left panel) or active construction (right panel), given different cues. In this simulation, the model is given a 2AFC task, so the probability of selecting a double object construction in the left panel and passive construction in the right panel is the complement of the displayed probabilities. Error bars are standard error derived from model performance from fifty resamples of the model's environmental vectors and prime-target pairs.

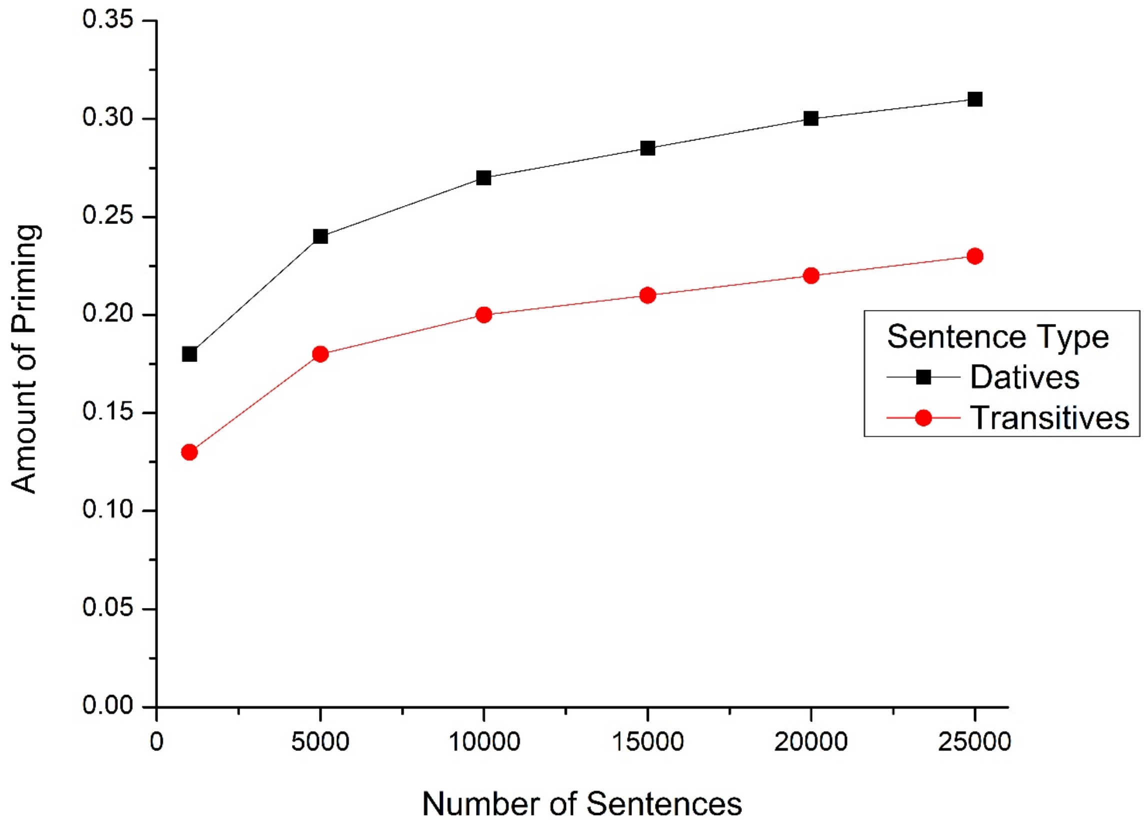

Figure 10. The impact of the number of sentences contained in memory on levels of priming for both dative and transitive sentences from Bock (1986).

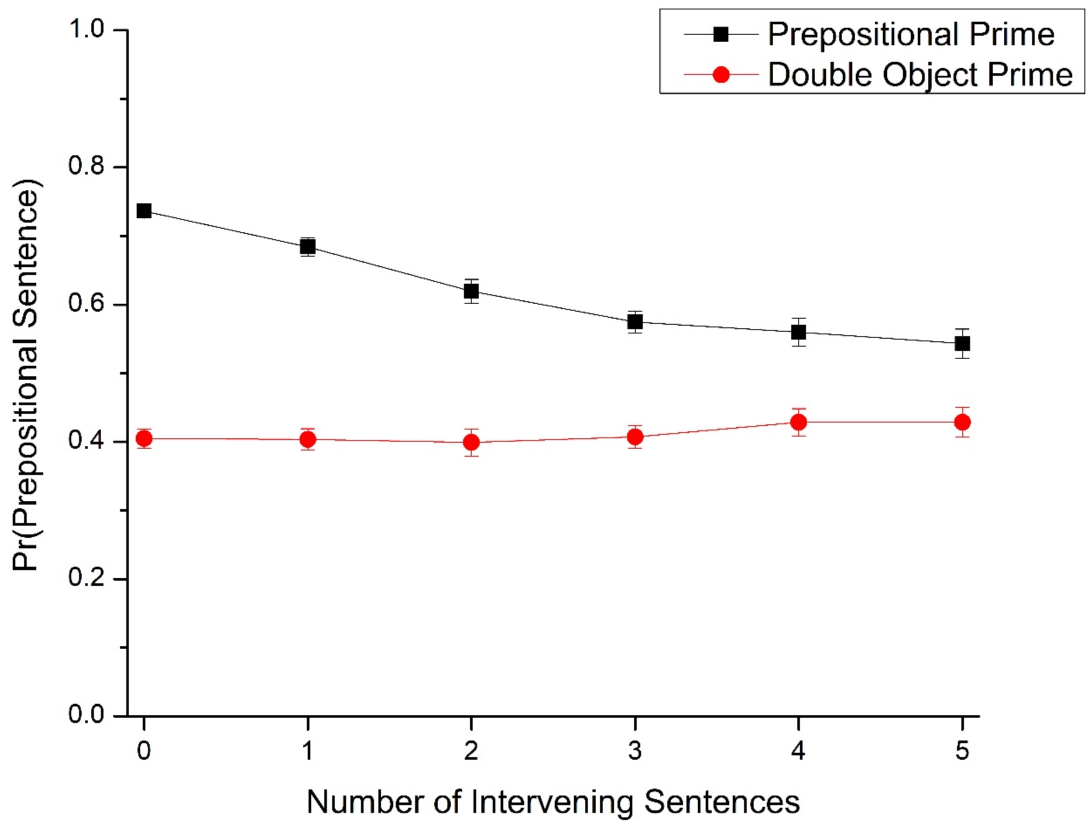

Figure 11. Simulation of the effect of intervening sentences on levels of syntactic priming in the IPM. Although the inclusion of intervening items in memory does decrease the size of priming, it is not eliminated. Error bars are standard error.

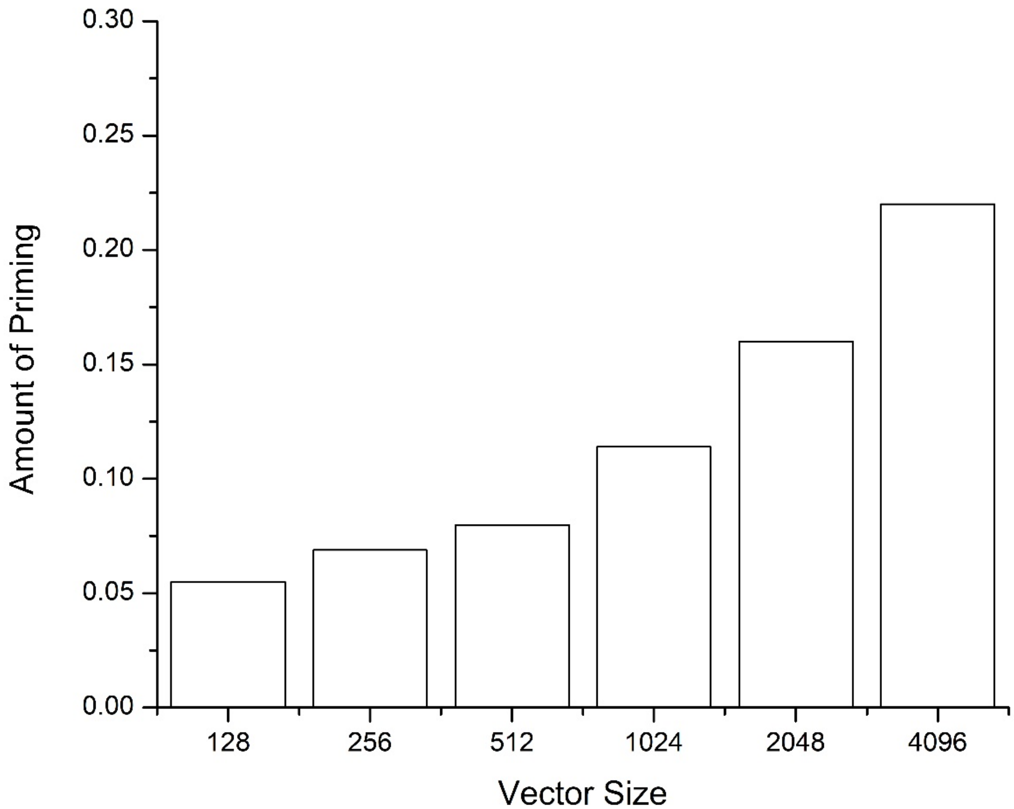

Figure 12. Effects of vector size on the amount of structural priming when there are 5 intervening sentences in memory. Larger vector sizes allow for more prime information to be retained in short-term memory, leading to larger priming effects, even with a significant amount of interfering information in memory.

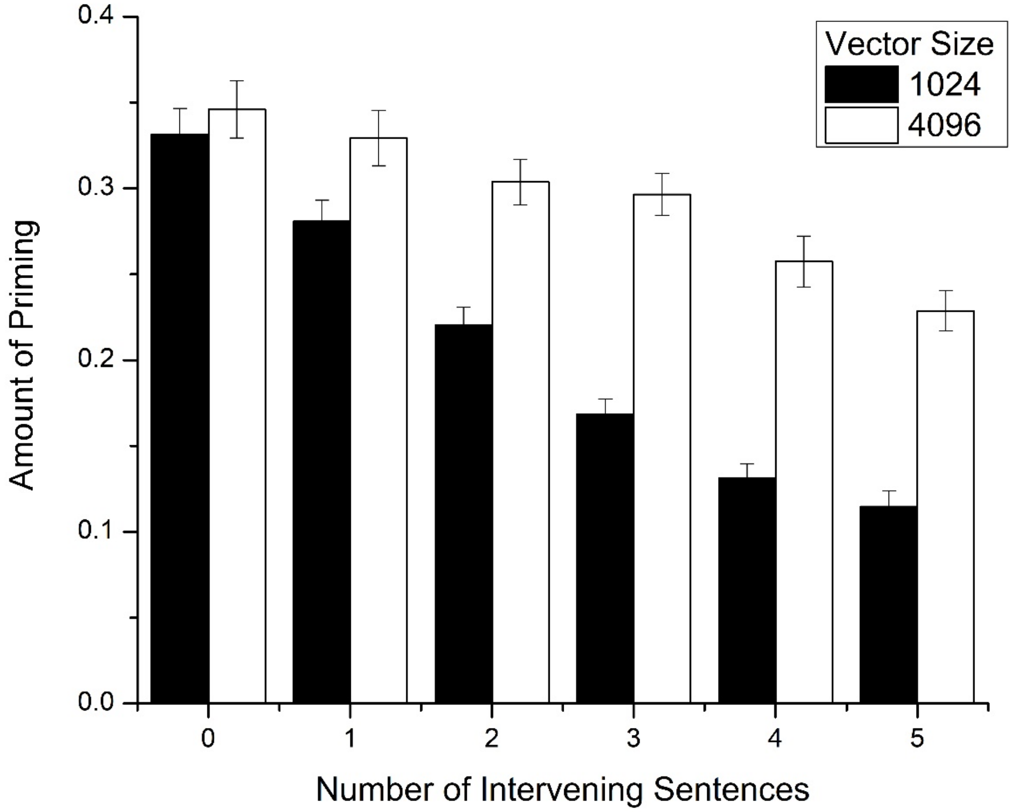

Figure 13. A comparison of levels of priming across number of intervening sentences for vector sizes of 1,024 and 4,096.

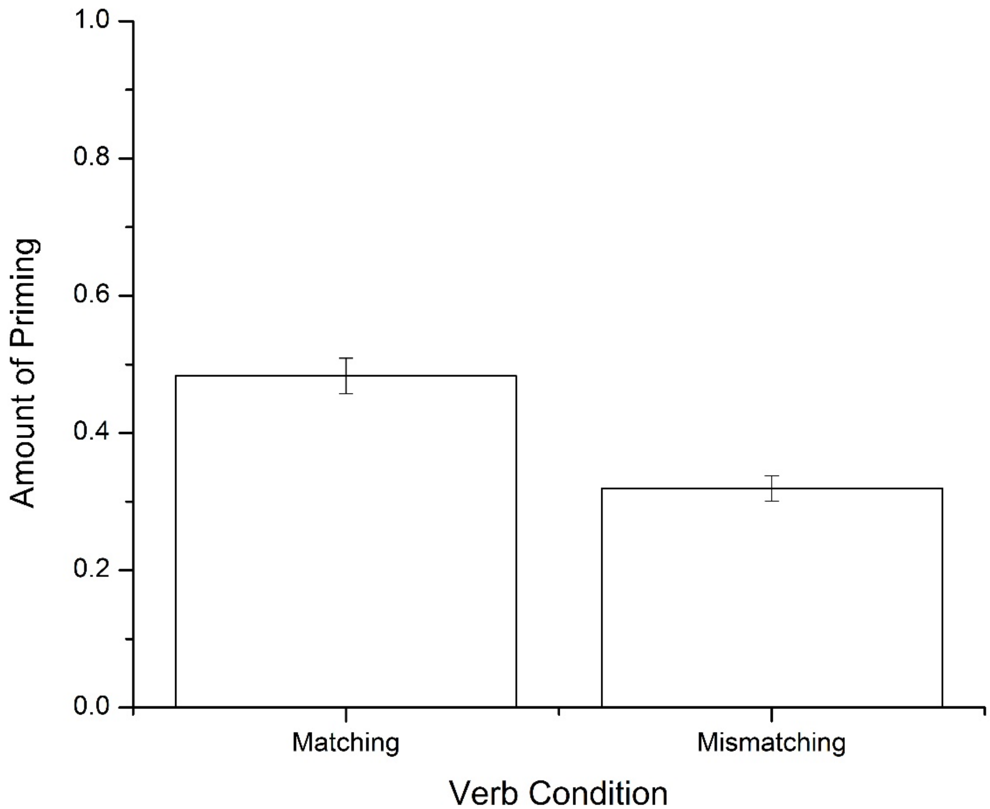

Figure 14. Simulation of the lexical boost in structural priming using sentences from Rowland, et al. (2012). This figure shows that there is a small but consistent increase in the level of priming when the prime has a matching verb with the target sentence.

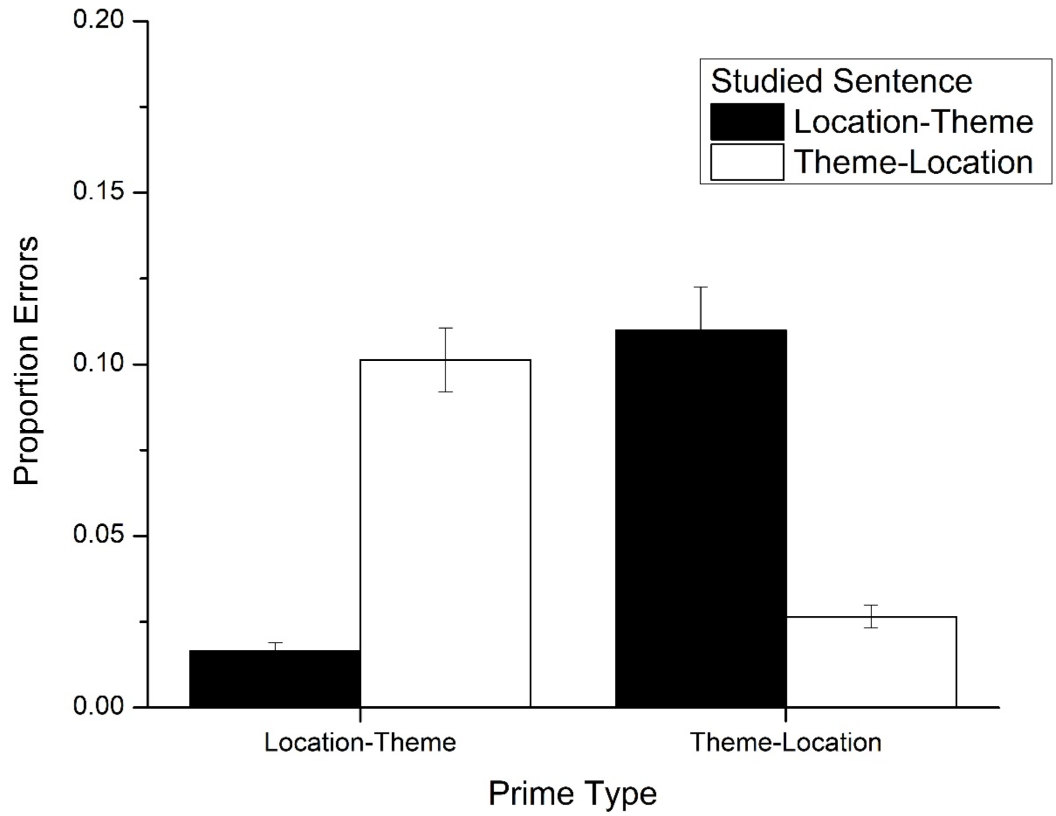

Figure 15. Simulation of the results from Chang, et al. (2003), demonstrating that the model can account for priming of within sentence orderings.