Controlling the retrieval of general vs specific semantic knowledge in the instance theory of semantic memory

Anonymous CogSci submission

Abstract

Distributional models of semantic cognition commonly make simplifying assumptions, such as representing word co-occurrence structure by prototype-like high-dimensional semantic vectors, and limit how retrieval processes may contribute to the construction and use of semantic knowledge. More recently, the instance theory of semantics (ITS, Jamieson, Avery, Johns, & Jones, 2018) reconceived a distributional model in terms of instance-based memory, allowing context-specific construction of semantic knowledge at the time of retrieval. By simulation, we show that additional encoding and retrieval operations, consistent with learning and memory theory, can play a crucial role in flexibly controlling the construction of general versus specific semantic knowledge. We argue this consolidation of processing principles holds insight for distributional theories of semantic cognition.

Keywords: distributional semantics; higher-order similarity; instance theory; surprise-driven learning; retrieval

Introduction

People flexibly understand word meaning at multiple levels of similarity and in different contexts. For example, the word cat could be similar to subordinate (tiger), basic (dog) or superordinate (animal) category words; associates (house, hat, or Felix); alternative meanings (a person in jazz, or Caterpillar machinery); or words in general compared to nonwords. How people flexibly control the specificity of their semantic knowledge remains unclear. Stable semantic representations for words may become positioned in high-dimensional semantic space to permit multiple comparisons for meaning along general and specific lines. Or, semantic representations could be more labile, potentially granting processes associated with encoding and retrieval operations control over the construction of general versus specific semantic meaning. The present work merges assumptions from three memory models, MINERVA 2 (Hintzman, 1984), the instance theory of semantics (ITS, Jamieson et al., 2018) and the instance theory of associative learning (MINERVA-AL, Jamieson, Crump, & Hannah, 2012), to examine how learning and memory operations participate in the flexible construction of general and specific semantic knowledge.

As accounts of semantic cognition, we distinguish between representation and retrieval models. Distributional models typically articulate processes for forming semantic representations but underspecify the role of retrieval. For example, models like LSA (Landauer & Dumais, 1997), BEAGLE (Jones & Mewhort, 2007), and word2vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) represent word meaning in terms of high-dimensional vectors sensitive to co-occurrence structure in natural text, whereby words closer in semantic space are closer in meaning. Furthermore, representations are prototypic because each word has one vector that roughly averages over co-occurrence structure with other words in the corpus. The prototype assumption is obvious for polysemous words. For example, "bank" could refer to a river or financial institution; but, a prototype representation averages the distinction with a single vector partway between the two meanings. Nevertheless, prototype representations can be sensitive to multiple levels of similarity (e.g., cat can be similar to lion and animal) because words can be positioned in high-dimensional space to somewhat align with multiple levels of meaning. Retrieval minimally involves comparing word similarity in high-dimensional space, but does not interact with encoding or construction of semantic representations. Finally, models use various pre-processing steps, like stop-word exclusion, log and entropy transformations (LSA), and negative information sub-sampling (word2vec), or post-processing transformations of base-rate information (e.g., Johns, Mewhort, & Jones, 2019) to further improve the specificity of semantic representations.

Outside of semantic modeling, instance/exemplar theories have risen against prototype accounts (e.g., in categorization and concept formation, Jacoby & Brooks, 1984). Instance theory assumes that memory encodes a history of richly featured examples as traces, and retrieves them in context-specific fashion by their similarity to patterns in the immediate environment. Although the composition of examples in memory is stable, knowledge as content retrieved from memory is labile depending on retrieval conditions and operations.

Recently, ITS (Jamieson et al., 2012) applied instance theory to distributional semantics by combining BEAGLE word representations (Jones & Mewhort, 2007) with MINERVA 2 encoding and retrieval operations (Hintzman, 1984). In BEAGLE, a semantic vector is a prototype aggregating over sentence representations containing a word. By contrast, ITS encodes individual sentences as memory traces, and aggregates over them at retrieval allowing retrieval conditions to selectively modulate the construction of word meaning. For example, ITS handles polysemy by retrieving the meaning of a probe word (bank) depending on the conditions of local context (river vs. piggy). ITS showed how the selective con-

struction of semantic knowledge from memory can depend on retrieval *conditions* (e.g., probe context), but did not fully consider how its encoding and retrieval *operations* offer additional control over the specificity of semantic representations.

Here, we establish the value of importing assumptions about encoding and retrieval operations from learning and memory theory into a modification of ITS (ITS 2). A parsimonious feature of ITS 2 is the consolidation of processing assumptions made between variants of MINERVA 2 that were not originally expressed in ITS. For example, MINERVA-AL (Jamieson et al., 2012) is an instance account of associative learning phenomena that employed a modified encoding rule, termed *discrepancy encoding*. Whereas MINERVA 2 and ITS encode each new experience as a raw trace in memory, MINERVA-AL mimicked the principle of surprise-driven learning (Rescorla & Wagner, 1972) by encoding only features of a new experience that were unexpected by memory. In ITS 2, we show that a form of *discrepancy encoding*, termed *weighted expectancy subtraction* (because it can be performed at encoding or retrieval) controls the specificity of semantic knowledge. Similarly, MINERVA 2 allowed the possibility of *iterative retrieval*, where memory responses inspire successive waves of retrieval. We show that iterative retrieval in ITS 2 allows traversal of higher orders of semantic similarity and controls the generality of semantic knowledge. In the general discussion, we speculate that encoding and retrieval operations are crucial for negotiating the integration of general and specific expectations for word co-occurrence in semantic representations, and may approximate post-processing transforms for weighting word base rate information known to improve word-embedding quality (Johns et al., 2019).

ITS and ITS 2

To overview we first define ITS and ITS 2, and then trained them on an artificial language with known word co-occurrence structure. This enabled clear accounting of encoding and retrieval operations controlled recovery of specific and general aspects of the semantic space.

Word representation Following BEAGLE, words are arbitrary perceptual objects with no pre-existing similarity. Each word, is assigned an environment vector, e_i, by randomly sampling n values from a normal distribution ($\mu = 0$, $\sigma = 1/n$), where n determines the dimensionality of the vector space. Thus, all words are ortho-normal in expectation. ITS can accommodate other representational assumptions and we used a identity matrix, with the diagonal set to 1, and the number of rows/columns equal to the number of words in the language.

Memory ITS preserves experiences with individual sentences in memory. For example, committing a sentence to memory involves summing the environmental vectors for the words in the sentence, and entering the composite vector as a new row in the matrix:

$$M_i = c_i = \sum_{j=1}^{j=h} e_{ij} \tag{1}$$

M is the memory matrix, and c_i is a sentence context. c_i is stored in a new row in M_i as a composite trace by summing the e_{ij} environment vectors for each word, from 1, to h, in the sentence. For example, the sentence context, c_i, "I like cats" is the sum of $e_I + e_{like} + e_{cats}$ word environment vectors. The number of words inside a trace, h, is a windowing parameter that must be larger than one word, otherwise the memory will return perceptually similar traces, rather than semantically similar ones. We note that the memory matrix becomes a document-term matrix of word frequencies when the environment vectors for words are taken from an identity matrix.

Retrieval Word meaning is constructed at retrieval. Memory is probed with a word and returns an echo response. The echo is the sum of similarity weighted traces to the probe, and taken as the semantic vector for the probe word. Retrieval and echo construction follow MINERVA 2. First, memory M is probed, p, with a word environment vector ($p_j = e_i$) and the cosine similarities between p_j and all traces M are computed to produce a vector of trace activations a_i:

$$a_i = \left(\frac{\sum_{j=1}^{j=n} p_j \times M_{ij}}{\sqrt{\sum_{j=1}^{j=n} p_j^2}\sqrt{\sum_{j=1}^{j=n} M_{ij}^2}}\right)^{\tau} \tag{2}$$

where, a_i is the activation (cosine similarity to probe) of trace i in memory, p_j are the jth features of the probe, M_{ij} are the jth features of each trace i in memory, and n is the number of columns in memory setting the dimensionality of the vector space. The vector of activations is raised to a power, τ, controlling a retrieval gradient determining selectivity in the composition of the echo. The activation vector is a record of similarity between the traces and the probe spanning the range -1 to 1, with $a_i = 1$ when a trace is identical to the probe, $a_i = 0$ when a trace is orthogonal to the probe, and $a_i = -1$ when the trace is opposite the probe.

Second, the memory-based semantic representation, m_i, for the probe word is retrieved as an echo by summing the traces in proportion to their activation. Specifically, all traces in memory are multiplied by their activations, and the echo is formed by summing the weighted traces:

$$m_i = echo_j = \sum_{i=1}^{i=m} \sum_{j=1}^{j=n} a_i \times M_{ij} \tag{3}$$

where, $echo_j$ is the jth feature of the echo, m is the number of traces in memory, a_i is the activation of trace i, and M_{ij} are the jth values of each trace i in memory. In ITS, the echo is used as the semantic representation for the probe word, m_i.

Words are compared for semantic similarity by comparing their respective echoes. Semantic similarity between two probes words, $cos(p_1, p_2)$, is computed between their respective echoes using a cosine:

$$cos(p_1, p_2) = \frac{\sum_{j=1}^{j=n} echo_{1j} \times echo_{2j}}{\sqrt{\sum_{j=1}^{j=n} echo_{1j}^2} \sqrt{\sum_{j=1}^{j=n} echo_{2j}^2}} \quad (4)$$

Briefly, words become similar to one another by appearing in similar sentences. For example, probing the word "doctor" will return an echo comprised of a sum over sentences including "doctor". This echo will be similar to the echo for words like "nurse" which sums over sentences with overlapping words (e.g., hospital).

ITS 2: Weighted expectancy subtraction at encoding

ITS 2 implements *weighted expectancy subtraction* during encoding in a similar manner to MINERVA-AL's *discrepancy encoding* rule. The difference is the subtraction between the probe and the echo is weighted by x, controlling the amount of expectation to be subtracted. Weighted expectancy subtraction is applied at each step across training. For example, when a new sentence is experienced, the sentence context vector c_i is used as a probe to memory to generate an echo. The echo represents the memories' expectation for the new sentence. If the new sentence is fully expected, then the memory can reconstruct the new sentence on the basis of its existing traces. The magnitude of the echo vector contains the sum of many traces, and is generally much larger than the magnitude of the sentence context vector. As a result, before subtraction, the probe and echo vectors are normalized,

$$c'_j = \frac{c_j}{\max |c_{j,n}|} \quad (5)$$

where, c_j is a sentence context probe vector, and the elements of c_j are divided by the largest absolute value in c_j, to produce the normalized c'_j. Similarly, the echo is normalized such that,

$$echo'_j = \frac{echo_j}{\max |echo_{j,n}|} \quad (6)$$

where, $echo_j$ is an echo vector, and the elements of $echo_j$ are divided by the largest absolute value in $echo_j$, to produce the normalized $echo'_j$.

Next, the new trace encoded to memory is defined by subtraction of a weighted normalized echo from the normalized probe,

$$M_{ij} = c'_j - x \cdot echo'_j \quad (7)$$

where, M_{ij} is the new row entry in the memory matrix, and x is a weighting parameter (from 0 to 1), controlling how the proportion of the normalized echo subtracted from the normalized probe. When x is set to 0, ITS 2 becomes equivalent to ITS.

ITS 2: Weighted expectancy subtraction at retrieval

ITS 2 can conduct *weighted expectancy subtraction* at retrieval, after training is complete. Memory is constructed identically to ITS, except weighted expectancy subtraction occurs at retrieval through a two-step *iterative retrieval* process. A probe word generates an echo from memory, and the echo is submitted as an "internal" probe to generate a second echo. The semantic representation for the word is taken as a weighted subtraction of the normalized second echo from the normalized first echo.

The first echo, $echo_\alpha$, is generated in the usual way, but then resubmitted as a probe to construct a second echo, $echo_\beta$, by the same equations 2 and 3 used to construct $echo_\alpha$. Both $echo_\alpha$ and $echo_\beta$ are normalized following equation 6. The semantic representation for a word, m_i, with weighted expectancy subtraction at retrieval in ITS 2 is:

$$m_i = echo'_\alpha - x \cdot echo'_\beta \quad (8)$$

where, m_i is the semantic representation for the ith word, and x is a weighting parameter varying from 0 to 1 controlling the proportion of $echo'_\beta$ subtracted from $echo'_\alpha$.

Simulations

Our aim was to characterize how ITS and ITS 2 recover specific and general aspects of semantics from co-occurrence. First, we created an artificial language with known co-occurrence structure. Next, we trained ITS on sentences from the artificial language and compared the semantic structure of ITS vectors to direct measures of the semantic structure of the language. We were interested in determining which aspects of the language ITS recovers by default. Last, we show that encoding and retrieval operations in ITS 2 provide control over the specificity of semantic knowledge production.

Artificial language

The artificial language contained no grammar and only semantic structure based on word co-occurrence. The simplistic form offers a transparent window into the transformations of ITS 2. We created semantic topic generators that use unique collections of words to discuss a given topic, with some overlap across topics. The language contained 100 words and 10 topics. Each topic used 15 words, and overlapped with neighboring topics by five words on both sides. Each topic had a random word-occurrence probability distribution that summed to one. Figure 1 depicts the topic-word probability matrix defining the artificial language. A corpus was generated by randomly sampling topics (equal probability), and then constructing sentences from the topic by sampling n words as a function of their probability. Sentence-size varied randomly between 10 and 20 words per sentence. A corpus included 5,000 sentences.

The purpose of the simulations was to compare the semantic spaces generated by ITS and ITS 2 to known properties of the semantic space from the language. We defined the known semantic space at various orders of semantic similarity. At the first order, the true semantic representation for a word was the column vector for each word in the topic-word probability matrix above. To visualize this semantic space we computed the cosine similarity between each word

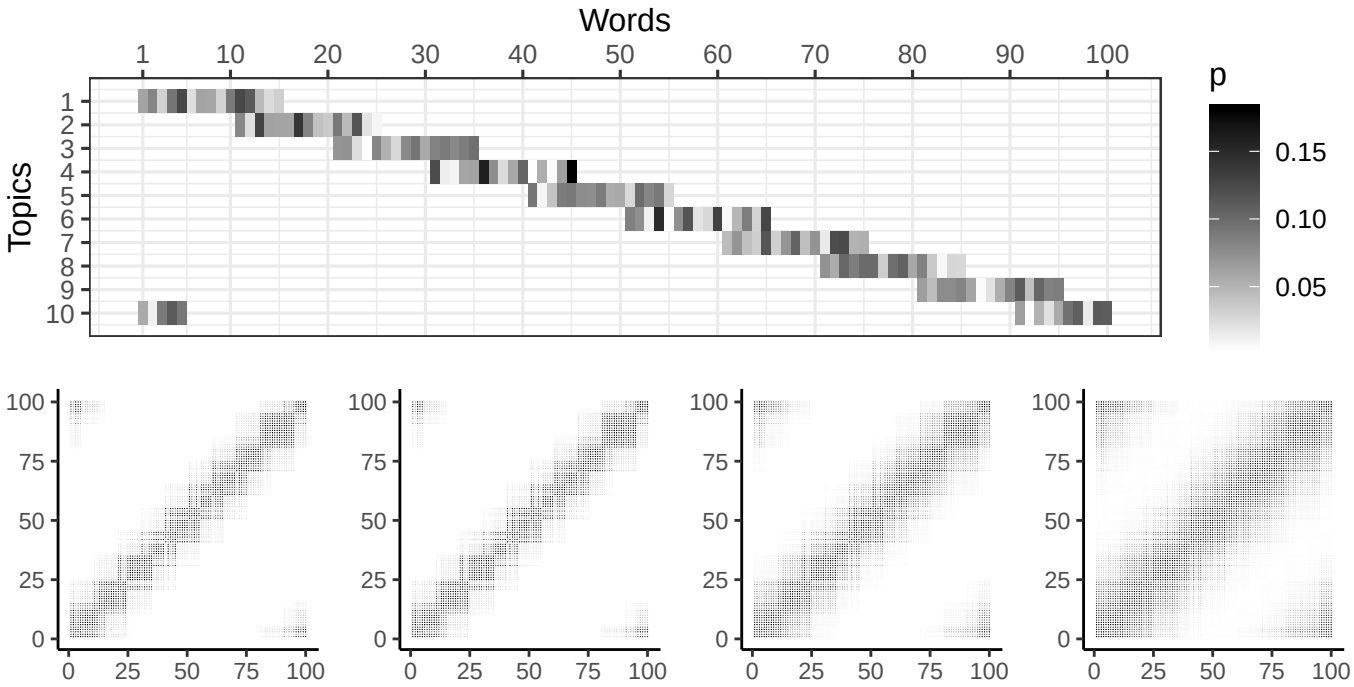

Figure 1: Upper: The topic-word probability matrix defining the artificial language. Darker colors represent higher probability of word occurrence. Lower: Word-word similarity matrices from the first to fourth order.

(using their column vectors) and plotted the similarity matrix. The first word-word similarity matrix in figure 1 (lower panel) shows the structure of the artificial language that models are ostensibly attempting to recover. Words are more similar to each other within their topics than between topics, and there is some overlap because word usage overlaps across the topics. Words in topic one are not at all similar to words in topic six because there is no overlap in word usage between those topics. The remaining panels in figure 1 show word-word similarity in higher order space up to the fourth order, reflecting more general semantic similarities between words. A higher order similarity space uses a lower-order space to derive a higher order one. For example, the second-order space uses columns from the first-order similarity matrix as word embeddings to compute a second word-word similarity space, and so on. In our language, because of word overlap between topics, words become increasingly similar to one another in higher order space. A veridical model would recover specific word meaning from first-order semantic space; whereas, more general word meaning could be recovered by accessing higher semantic space.

Simulation 1: ITS

We trained ITS on 5000 sentences, using one-hot coding (100×100 identity matrix) to form environment vectors for the words. Each word was coded as a 1, with 99 zeroes. The position of the 1 in the vector refers to the *nth* word in the corpus. As a result, the memory matrix is equivalent to a document-term matrix of raw term frequencies occurring in each document. We used a range of retrieval gradients (τ =

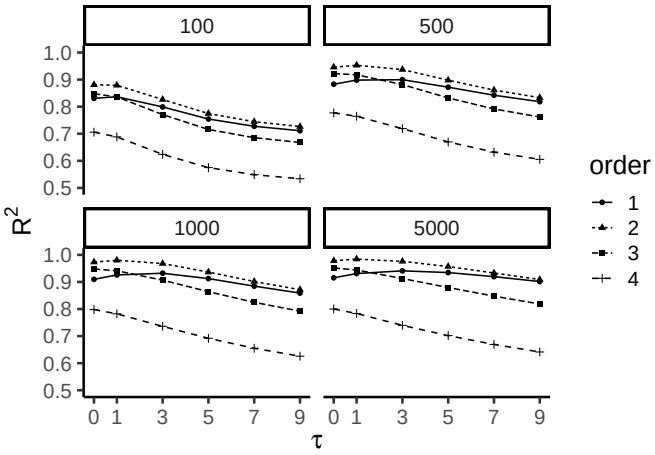

Figure 2: R^2 values between ITS word-word similarity space, and the first to fourth order word-word similarity spaces derived the artificial language as a function of training, and retrieval gradient (τ)

0 to 9) and training intervals (100, 500, 1000, and 5000 sentences). At each interval we computed echoes as semantic representations for each word, and then a word-word similarity matrix from those vectors. To determine which aspects of the artificial language ITS recovered, we computed R^2 between the ITS word-word similarity space, and the first to fourth order word-word similarity spaces derived directly from artificial language. The results are shown in figure 2.

ITS performed well in recovering the structure of the lan-

guage, and *was most sensitive to second-order similarity space*. Overall, ITS became more sensitive to all orders of similarity as training increased, and less sensitive as τ increased. Raising τ did increase relative sensitivity to the first order, but did so at the cost of losing sensitivity overall. Specifically, increasing the selectivity of the retrieval gradient limits the sampling of traces into the echo, resulting in noisier representations. The fact that ITS prioritizes the second order over the first is a flaw. The second order space is an overgeneralized version of the first, and blurs out the finer distinctions between word usage within the topic structures that generate the words. ITS relies on second order similarity (see discussion), so semantic vectors for topic-unique words become similar to words from overlapping topics, whereas they are not similar to those words in first order space. ITS glosses over these nuances.

Simulation 2: ITS 2 encoding

We next trained ITS 2 with *weighted expectancy subtraction* at encoding on the same artificial language. We show that weighted expectancy subtraction causes ITS 2 to become more sensitive to first order word-word similarity than higher orders. In the simulations we vary the value of x (from .01 to .5) to subtract different amounts of the echo from the probe. The value of x causes systematic differences in ITS 2's sensitivity to higher order similarity structure. For brevity, we report results with τ set to 1 (shown in figure 3, left panel).

Weighted expectancy subtraction at encoding modulated how ITS 2 recovered different orders of semantic similarity space, specifically allowing recovery of more veridical and nuanced word embeddings from the first-order similarity space. For example, when $x = .01$, ITS 2 was most sensitive to second order similarity, but as x increased ITS 2 became most sensitive to first-order similarity. Increasing x further caused overall sensitivity to decline.

Simulation 3: ITS 2 retrieval

Here, we repeated the above simulation but applied *weighted expectancy subtraction* with *iterative retrieval* after training was complete (using standard ITS memory encoding). The results are shown in figure 3 (right panel).

Remarkably, ITS 2 does not need to make any assumptions about encoding to benefit from *weighted expectancy subtraction*. The pattern of Simulation 3 is almost identical to that of Simulation 2. Specifically, ITS 2 becomes most sensitive to first-order word-word similarity structure as x is increased. Again, increasing x has diminishing returns.

General Discussion

We showed that ITS is most sensitive to second order semantic space, and that ITS 2 increases sensitivity to the more veridical first order space by processes of *weighted expectancy subtraction* and *iterative retrieval*.

It is instructive to consider how ITS and ITS 2 recover different orders of similarity space. First, consider how words become increasingly similar across orders of similarity space. In the first order, word similarity is determined by the topics they occur in. Word 6 is unique to topic one and only similar to words in topic one. In the second order, words become similar on the basis of their first-order similarity features. First-order features for word 6 contain positive similarity for topic one words 1 to 15. Some of these features (11 to 15) are shared by words from topic two, so word 6 becomes similar to topic two words in second order space. If topics are connected by overlapping words, then all words become increasingly similar across increasing orders of similarity, and the nth order similarity matrix becomes all ones.

Crucially, *iterative retrieval* in ITS 2 is a process for traversing higher-order similarity space; and *weighted expectancy subtraction* is a process for negotiating the relative contributions of higher-order similarity in the construction of semantic knowledge. To elaborate, we showed that standard ITS echoes are most sensitive to the second order. Echoes contain sentence memory, so an echo for a topic-unique word is immediately partially similar to echoes for words from neighboring topics, because their echoes share co-occurring words. Submitting an echo as a probe for iterative retrieval is a third order operation. The echo contains many words and the second echo collapses over memory for sentences that contain any of those words. This draws in sentences from additional topics, causing a given word to be more similar to words in more distant topics. Iterating to the extreme sweeps all sentences in memory into the echo, causing identical echoes for all words.

Simulation 3 showed that subtracting a portion of the second echo from the first allows ITS 2 to preferentially recover first order space. Our preceding discussion suggests ITS 2 performs a weighted subtraction of third from second order space, implying a similar result could be obtained analytically. We confirmed this directly from the language by subtracting proportions of the third order similarity matrix from the second, and computing R^2 between each new matrix and the first order similarity matrix. We found an inverse U function, with R^2 approaching 1 at .4. As a sidenote, computing second order similarity from a document term matrix (Cribbin, 2011) can produce embeddings similar to those produced by singular value decomposition, as in LSA (Landauer & Dumais, 1997). We speculate that subtracting a portion of the third order from the second may further improve the quality of those semantic representations.

More generally, count-based/vector-accumulation models like ITS rely exclusively on positive information from word co-occurrence, whereas neural embedding models like word2vec (Mikolov et al., 2013) exploit negative information by sub-sampling adversarial examples during training which may result in superior word embeddings (Mandera, Keuleers, & Brysbaert, 2017). Johns et al. (2019) developed analytic transformations for weighting word occurrence base rates that approximate gains from using negative information for improving word-embedding quality. We speculate that ITS 2 negotiates a similar merger of general expectations for word

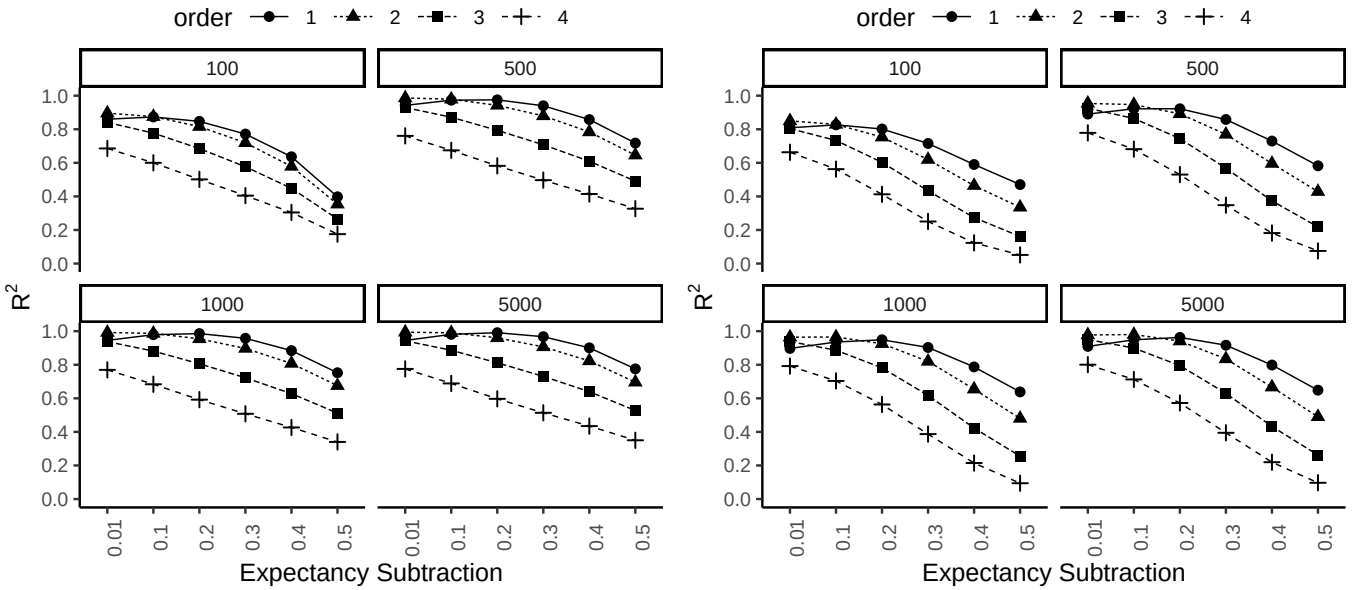

Figure 3: R^2 values between ITS word-word similarity space and the first to fourth order word-word similarity spaces derived the artificial language as a function of training, and *weighted expectancy subtraction*. The left panel shows ITS 2 with weighted expectancy subtraction during encoding, and the the right panel shows ITS 2 with weighted expectancy subtraction during retrieval.

occurrence in higher order space with specific expectations from lower order space by *iterative retrieval* and *weighted expectancy subtraction*; and, may realize base-rate transforms through cognitive encoding and retrieval operations.

As a future direction we will apply ITS and ITS 2 to natural language and determine whether ITS 2 assumptions produce higher quality fits to human semantic judgments. At present, we offer ITS 2 as an intriguing account of how people may transform their semantic knowledge along general versus specific lines, by using *iterative retrieval* to traverse higher order similarity space, and *weighted expectancy subtraction* to control the specificity of retrieved semantic knowledge.

References

Cribbin, T. (2011). Discovering latent topical structure by second-order similarity analysis. *Journal of the American Society for Information Science and Technology*, *62*, 1188–1207.

Hintzman, D. L. (1984). MINERVA 2: A simulation model of human memory. *Behavior Research Methods, Instruments, & Computers*, *16*, 96–101.

Jacoby, L. L., & Brooks, L. R. (1984). Nonanalytic cognition: Memory, perception, and concept learning. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 18, pp. 1–47).

Jamieson, R. K., Avery, J. E., Johns, B. T., & Jones, M. N. (2018). An instance theory of semantic memory. *Computational Brain & Behavior*, *1*, 119–136.

Jamieson, R. K., Crump, M. J. C., & Hannah, S. D. (2012). An instance theory of associative learning. *Learning & Behavior*, *40*, 61–82.

Johns, B. T., Mewhort, D. J., & Jones, M. N. (2019). The Role of Negative Information in Distributional Semantic Learning. *Cognitive Science*, *43*, e12730.

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

Landauer, T. K., & Dumais, S. T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, *104*, 211–240.

Mandera, P., Keuleers, E., & Brysbaert, M. (2017). Explaining human performance in psycholinguistic tasks with models of semantic similarity based on prediction and counting: A review and empirical validation. *Journal of Memory and Language*, *92*, 57–78.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black & W. F. Prokasy (Eds.), *Classical conditioning II: Current research and theory* (pp. 64–99). New York: Appleton-Century Crofts.