Comparing Predictive and Co-occurrence Based Models of Lexical Semantics Trained on Child-directed Speech

Fatemeh Torabi Asr
Cognitive Science Program
Indiana University, Bloomington
fatorabi@indiana.edu

Jon A. Willits
Department of Psychology
University of California, Riverside
jon.willits@ucr.edu

Michael N. Jones
Psychological and Brain Sciences
Indiana University, Bloomington
jonesmn@indiana.edu

Abstract

Distributional Semantic Models have been successful at predicting many semantic behaviors. The aim of this paper is to compare two major classes of these models – co-occurrence-based models, and prediction error-driven models – in learning semantic categories from child-directed speech. Co-occurrence models have gained more attention in cognitive research, while research from computational linguistics on big datasets has found more success with prediction-based models. We explore differences between these types of lexical semantic models (as representatives of Hebbian vs. reinforcement learning mechanisms, respectively) within a more cognitively relevant context: the acquisition of semantic categories (e.g., *apple* and *orange* as fruit vs. *soap* and *shampoo* as bathroom items) from linguistic data available to children. We found that models that perform some form of abstraction outperform those that do not, and that co-occurrence-based abstraction models performed the best. However, different models excel at different categories, providing evidence for complementary learning systems.

Introduction

Distributional models of lexical semantics have had a large impact on cognitive science over the past two decades. In general, these models formalize the distributional hypothesis (Firth, 1957; Harris, 1970), and attempt to learn distributed representations for word meanings from statistical regularities across a large corpus of linguistic input. The resulting representations have been enormously valuable to researchers wanting to select and calibrate word stimuli balanced on semantic dimensions. They have also been successfully used as semantic representations in models of cognitive processes (e.g., word recognition, reading), and in a wide variety of applications ranging from automated tutoring to open question answering.

Due in part to their practical successes, the algorithms that distributional models use to build semantic representations have also been hypothesized to be related to the cognitive mechanisms potentially used by humans to learn semantic representations from regularities in their language input[1]. The various learning mechanisms posited include simple co-occurrence learning, episodic abstraction, reinforcement learning, and probabilistic inference (see Jones, Willits, & Dennis, 2015, for a review).

The majority of distributional models in the cognitive science literature are from the family of *co-occurrence models*. Models of this family tend to apply unsupervised learning mechanisms to a frequency count of how often words co-occur with each other in a context (paragraph, document, or an *n*-word moving window). There are many specific models that differ in theoretically meaningful ways in terms of the learning mechanisms they apply, but all members of this family share the assumption that the learner is basically counting observed co-occurrences of stimuli in the environment. Hence, they are all based on error-less Hebbian-type learning mechanisms.

Perhaps the best-known co-occurrence model in the cognitive literature is Latent Semantic Analysis (LSA; Landauer & Dumais, 1997). LSA basically applies a dimensionality reduction mechanism to a sparse word-by-document frequency matrix computed from a large corpus of text. The resulting dense vectors emphasize higher-order statistical relationships among words: Two words that occur in similar contexts across language will have similar representations, even if they never directly co-occurred in a context. LSA vectors have been used to model how words' semantic similarity predicts performance in vocabulary acquisition, categorization, reading, summarization, and a number of other cognitive tasks (see Landauer, 2006, for a review). Other popular models count the co-occurrences of words within a moving window, or how frequently words occur with predefined context words. For a review of the various co-occurrence algorithms and their performance on semantic tasks, we refer the reader to Bullinaria and Levy (2007) or Riordan and Jones (2011).

A second family of *predictive* distributional models has been around for decades, based on principles of predictive encoding and error-driven learning core to theories of reinforcement learning. For example, early recurrent neural networks studied by Elman (1990) and St. John and McClelland (1990) learn distributed representations for a word's co-occurrence history across their hidden layers. Feedforward networks studied by Rogers and McClelland (2004) similarly learn distributed semantic representations, representations that even contain hierarchical taxonomic relations. These model architectures rely on predicting a distributed set of features for each input word, then deriving an error signal from the difference between the prediction

[1] To be fair, the transfer between practical algorithm and human mechanism has been bidirectional: Knowledge of how humans learn was used to inform the design of distributional algorithms, and the subsequent success of certain algorithms on practical tasks has then fed back to help narrow the range of likely cognitive mechanisms that humans use.

and the observed context. Errors are backpropagated through the network to increase the likelihood that the correct output will be predicted given the input in the future.

The mechanisms for predictive learning core to these models have been studied a great deal within reinforcement learning, and there is a considerable literature in cognitive neuroscience exploring how this type of learning is driven at the neural level by dopaminergic error signals, and is moderated by the basal ganglia, cingulate, and hippocampus. In practice, these predictive models have only been applied to small toy datasets to discover distributional structure; their capabilities have not been studied when applied to large, naturalistic corpus-based materials such as those that are used by simple co-occurrence models. However, there has been a recent resurge of interest in predictive models of distributional semantics. Howard et al., (2011) trained a predictive version of the Temporal Context Model (pTCM), a recurrent model of error-driven hippocampal learning, on a large text corpus and demonstrate impressive performance on word association tasks. Similarly, Shaoul et al. (2016) used a Naïve Discrimination Learning (NDL) procedure (a single-layer network trained using the Rescorla-Wagner learning procedure) to show that semantic representations can be learned from a relatively small sample of spoken language. Despite the many appealing properties of predictive distributional models, the bottom line is that they are heavily outperformed by co-occurrence models at accounting for human data on every lexical semantic task that has been tested.

A new type of context-prediction model (a.k.a. "neural embedding model") has emerged in the past few years; as Baroni et al. (2014) put it, these models are "the new kids on the distributional semantics block." The predictive model of Mikolov et al. (2013), referred to in the literature as *word2vec* (W2V), is a feedforward neural network model with a hidden layer that uses error backpropagation to maximize the likelihood of either predicting context given a word, or predicting a word given the context. In this sense, W2V behaves much like the networks studied by Rogers and McClelland (2004), but applied to massive amount of linguistic data, and with some tricks to improve training efficiency. W2V has made a huge stir in the machine learning literature for its ability to outperform every other semantic model on benchmark tasks, and to achieve this impressive performance using an architecture that had been written off in the cognitive and linguistic literatures. Baroni et al. (2014) undertook a careful comparison of state-of-the-art co-occurrence models and W2V, testing them on the same input corpus and with a large battery of different semantic tasks. They concluded that the hype surrounding W2V is warranted: Even under these very well controlled comparisons, W2V outperformed the current top performers studied by Bullinaria and Levy (2007). Since W2V has a similar architecture to many "toy" connectionist models that have been popular in cognitive science, its success on practical tasks is exciting to the field. However, it is important to note that current tests of W2V have been on very large corpora—the tests by Mikolov et al. (2013) were trained on over 1-billion words of text, and the comparisons by Baroni et al. (2014) were trained on a corpus of almost 3-billion words. While estimates of the number of tokens a human will read/hear in a lifetime vary greatly, both of those corpora are orders of magnitude beyond the upper limit of a single human's experience.

Hence, our approach here is to scale the problem way down. We train co-occurrence and predictive models on the real-world speech that children experience from birth to age 5 using the CHILDES corpus (MacWhinney, 1998), a resource used in previous work on computational modeling of early word categorization (e.g., Asr et al., 2013). This linguistic experience is a very different test for the models, and allows us to explore how their learning mechanisms might deal with the noisy data on which children must build their semantic representations. Given the superiority of W2V over co-occurrence models on large data and practical semantic tasks, and the similarity of its learning algorithm to popular error-driven models of development, how do the two families of models compare on their ability to learn complex structure on the same impoverished data that children receive?

Co-occurrence Models

To evaluate the performance of co-occurrence based models, two models were selected that have both performed well in previous evaluations, but that use different learning algorithms (thus providing breadth of coverage of different types of co-occurrence based models). One model uses an abstraction mechanism, whereas the other operates with simple summation of surface-level word co-occurrences.

PCA-Based Vector Model

The first model tested is notable for its use of principle components analysis (PCA) as the primary method of knowledge abstraction. This model computed co-occurrences in a 12-word moving window (12 words in both the forward and backwards directions) for the 10,000 most frequent words in the corpus, resulting in a 10,000-by-10,000 co-occurrence count matrix. These values were then normalized into positive point-wise mutual information values (Bullinaria & Levy, 2007). This matrix was then reduced using PCA, and the first 30 principle components were retained, resulting in 30-element vectors for each of the 10,000 words. This model is composite of several pre-existing models, such as the HAL model (Lund & Burgess, 1996), the COALS model (Rohde et al., 2005) and models by Bullinaria & Levy (2007).

Sparse Random Vector Accumulator

The second co-occurrence model we tested was from the family of Random Vector Accumulators (RVAs; see Jones, et al., 2015, for a review). RVAs are essentially distributed count models. They initialize a unique random vector for each word prior to learning, and the word's memory vector is then updated across learning as the sum of the vectors representing the words with which it has co-occurred.

Hence, RVAs are incremental learners with no abstraction mechanism (compared to PCA). We use a sparse-distributed RVA here (Recchia et al., 2015). To equate with the other models, the RVA was trained on the same 10k x 10k word co-occurrence matrix as was the PCA model. Each word was initially represented by an 8,000-element sparse ternary environment (E) vector. The memory (M) vector was then the frequency-weighted sum of the E vectors for all the words with which the target word co-occurred within a 12-word context window. For example, if *dog* had a co-occurrence frequency of 3, 1, 0, 2, with *cat, shoe, bunny*, and *run*, respectively, then $M_{dog} = 3*E_{cat} + 1*E_{shoe} + 2*E_{run}$.

Predictive Models

To compare to the RVA and PCA co-occurrence models, we tested two error-driven models. The primary objective was to evaluate the performance of the W2V predictive learning algorithm, but we also tested a second model of the error-correction family based on the classic Rescorla-Wagner model of discrimination learning.

W2V

As described above, W2V is a multilayer neural network (with an input layer, an output layer, and one hidden layer) that learns word vectors by iterating through sample contexts. W2V comes with two slightly different architectures of a neural network for learning word embeddings: (1) *cbow* (context bag-of-words) and (2) *skipgram*. In the cbow architecture, a word is predicted as the output from its context input. During training, the input layer produces a weighted sum over the context words within a fixed adjacency window of the target word. Output activations are converted to a probability distribution over the vocabulary (softmax) and the weight matrices are updated through backpropagation of the errors. The network topology is similar for skipgram, except that in this architecture, a target word is used to predict the context in which it appears. In skipgram, several context vectors can be sampled from a certain window of adjacent words (e.g., given the input sentence "she found a cute cat in the garden", the target word "cat" can be used to predict context unigrams "cute", "in", "a" and "the" when a window of size two (from each side) is considered. After training the model on a text corpus, the weight matrix between the hidden layer and the output layer in cbow or the one between the input layer and the hidden layer in skipgram represents the embeddings for the vocabulary words: V * N, where V is the size of vocabulary and N is the dimension of word vectors. In our experiments, we used a python implementation of the W2V model from Gensim (Řehůřek & Sojka, 2010).

Naïve Discrimination Learner

Like W2V, the Naïve Discrimination Learning (NDL) model learns to make predictions about a target word given its lexical context, or vice-versa. Unlike W2V, NDL has only an input layer and an output layer (with no hidden layer), and uses the Rescorla-Wagner learning procedure (Rescorla & Wagner, 1972) to learn a set of weights between target words and its cues (i.e. the other words in the window) that predict it. In addition to a long history of use in the psychology of learning (see Miller et al., 1995, for a review), NDL models are now being explored as models of infant word segmentation and how children learn the meanings of words (Baayen et al., 2015).

Method

Corpus

We used the entire child-directed speech data in the American English subset of the CHILDES corpus (MacWhinney, 1998). This collection includes conversations between children (4 to 60 months of age) and their parents, care-givers and other children. Utterances directed to the target children were combined to create a corpus representative of the linguistic input of children from these ages. The resulting corpus consisted of 4,568 sub-corpora (transcribed documents), containing 36,170 distinct word types and 8,323,266 total word tokens. The corpus is relatively well-distributed across ages and generally forms a decent snapshot of the input children get at various ages. Little pre-processing was done to the corpora beyond simple word tokenization. To equate comparisons across the models, from this corpus, we selected the 10,000 most frequently occurring words and used only those words as inputs into the four models. Words below this rank were excluded due to their low frequency in the corpus (<7).

Evaluation Task

We evaluated the performance of the models based on a word categorization task. For this task, we used 1,244 high frequency nouns from the corpus that unambiguously belonged to a set of 30 categories (like *mammal*, *clothing*, etc.). Thus, each non-identity pair of words either belonged to the same category (as in *dog-cat* for mammals and *shoe-sock* for clothing) or to different categories (as in *dog-shoe*, *dog-sock*, *cat-shoe*, and *cat-sock*).

The category membership of each pair was predicted using each model's similarity score in a signal detection framework. For each word pair, the similarity score was compared to a decision threshold. If the similarity score was above that threshold, the pair was predicted to belong to the same category (classified a "*hit*" if this prediction was correct and a "*false alarm*" if this prediction was wrong). If the similarity score was below the threshold, the pair was predicted to belong to different categories (classified a "*correct rejection*" if this prediction was correct and a "*miss*" if this prediction was wrong). For each model, a single decision threshold was chosen, the threshold that maximized categorization accuracy for that model. Each model's overall performance was assessed by computing balanced accuracy (BA) using the formula below:

$$BA = \frac{1}{2} * \left(\left(\frac{hits}{hits + misses}\right) + \left(\frac{correct\ rejections}{correct\ rejections + false\ alarms}\right)\right)$$

Experiment 1

Our first experiment focused on exploring the parameter space of the W2V model to see how an error-driven distributional model learns from the specific characteristics of child-directed speech data: short sentences, simple structure, less ambiguity, etc.

W2V Setup

The skipgram architecture is known to perform better on smaller corpora and modeling rare words due to repeated sampling from a fixed window of context words, whereas cbow is trained faster, thus is more suitable for learning word embeddings from big text corpora. We tried both architectures in this experiment to see if this held true in our task. We also manipulated two other parameters in the model: the window size for collecting context words, and the hidden layer size, i.e., dimensionality of the resulting word embeddings. We examined window sizes of 2 and 12 words from each side of the target word. Previous experiments showed that context words in a smaller window convey syntactic information about the target word, whereas, context words in a larger window convey more topical information (Levy & Goldberg, 2014a). Vector dimensionality (size of hidden layer) was set to 30, 50, 100, 200 or 300 nodes. Larger hidden layers provide a finer-grained distributional representation, and thus can be beneficial in learning word similarities to perform a categorization task. In contrast, smaller hidden layer sizes force the network to develop more abstract representations of meaning. Frequency cut-off was set to 1. Other parameters in the W2V training function were held constant across the experiments (for a complete description of the default parameter values please check the Gensim package).

Results

Figure 1 shows the accuracy of the model with different parameter considerations in the categorization task. As expected, the skipgram architecture learned similarity between the words better than the cbow architecture, resulting in superior categorization performance. Interestingly, considering a larger context window size (changing from 2 to 12 context words from each side of the target word) did not enhance the performance of the cbow model significantly. On the other hand, the skipgram model did benefit from more context words. When compared against the cbow results, this suggests that the size of our corpus is small for the connectionist model to learn the similarities between the words in our categorization dataset. That is why sampling smaller *n*-grams from a bigger window of adjacent words (the mechanism used only in the skipgram model) results in a significant boost in the backend performance. This observation is in line with previous experiments on large corpora (Mikolov et al. 2013) where the skipgram model outperformed cbow in semantic association tasks. The cbow model, on the other hand, recognizes syntactic associations (plural/singular) slightly better. This can be due to the way every training instance is used in either model: in skip-gram, unigrams are sampled from the context window and used one-by-one together with the target word in several training steps, whereas cbow uses the entire context frame of a word at once.

The second observation concerned the effect of vector dimensionality. While adding nodes to the hidden layer increased the categorization accuracy in lower scales (i.e., from 30 to 50) the effect disappeared in higher scales (100 to 300). In fact, the best performing model overall is the skipgram with 200 hidden layer nodes (accuracy = 74.9%), not the one with 300. This can be due to the small and relatively less ambiguous vocabulary in the child-directed corpus, for which a smaller hidden layer would suffice and might generalize better to capture similarities between words within the coarse-grained categorization defined in our evaluation task.

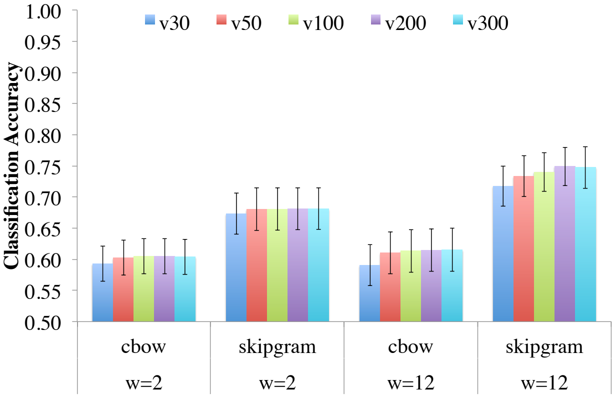

Figure 1. Mean classification accuracy (averaging across the 30 semantic categories, error bars represent 95% CI) of the various W2V models, as a function of the window size (2 vs. 12), the size of the hidden layer vector (30, 50, 100, 200, 300), and architecture (cbow, skipgram).

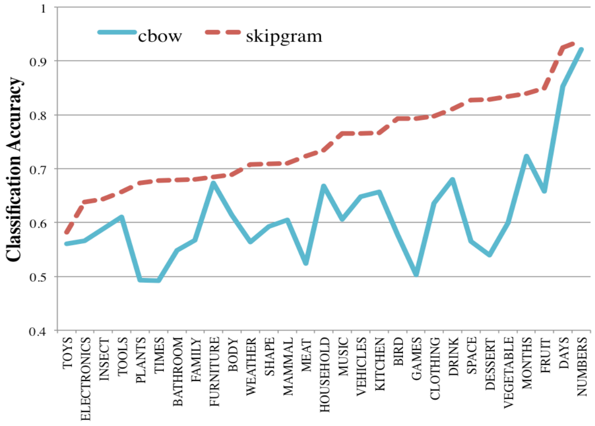

Figure 2. Mean classification accuracy on the 30 different categories, comparing the best performing cbow model to the best performing skipgram model (in both cases, w=12, v=200).

The best performing W2V setup will be used in our next experiment in comparison to other models. Thus before we proceed, it is good to take a closer look at the learning of each word category by either of the architectures. Figure 2, compares the best cbow and skipgram models on every category within the evaluation dataset. First, the two models seem to be learning qualitatively different types of information: cbow captured the within-category similarity of furniture and household items relatively better than it did on dessert, fruit, vegetable and meat items. In contrast, household and furniture were among the less accurate categories when detected by the skipgram model. Both models performed very well in categorization of numbers, days and months, but had difficulty with other categories such as plants. This suggests that the distributional properties of different semantic categories might be different. A correlation analysis of system performance across categories with *token frequency*, *word types* and *entropy* of the category items revealed that cbow learned categories with larger token frequency better (p<0.05), while the performance of skipgram on a category was statistically independent of these factors.

Experiment 2

In the second experiment we compare different distributional models in the word categorization task.

Setup of the Models

We trained each model with its optimal parameter settings. As we found in Experiment 1, for W2V this meant using the skipgram architecture with a window size of 12 and 200 nodes for the hidden layer. For comparison, the optimal performing NDL model had a window size of 3, and like W2V, performed better predicting contexts from words rather than vice-versa (Shaoul et al., 2016). NDL has no hidden layer, and therefore is not comparable in terms of vector size. Since the co-occurrence models do not use error-driven learning, direction (word predicting context vs. context predicting word) was not a feature of these models. However, these models were both trained using a window size of 12 (making them comparable to the better performance of W2V). The PCA model has a parameter equivalent to the number of hidden units in W2V (i.e. the amount of abstraction that is used): the number of principle components retained. The peak performance of this model was obtained with 30 principle components. The RVA model also has a parameter, which is the size of the random environment vectors. The peak performance of the RVA model was obtained with a random vector size of 8000.

Results

Figure 3 compares the performance of each of the four models in the word categorization task. The two co-occurrence models, PCA and RVA performed very differently. The PCA model had the advantage of learning latent relationships in the dataset, which likely helped it perform better on a small dataset such as the CHILDES corpus. Between the two predictive models, W2V was superior to the NDL. As with the PCA model, W2V's hidden layer allows it learn abstract, latent relationships from the co-occurrence data. In machine learning, introducing a hidden layer to a neural network topology is considered as a method for capturing nonlinear correlations between the input and output of the model. In addition to NDL missing the hidden layer, the input to this model was different from that of the W2V skipgram used in this experiment. In fact, like the cbow model tested in our previous experiment, NDL did not benefit from larger context windows (thus its peak performance is that of it being trained on 3 context words from each side).

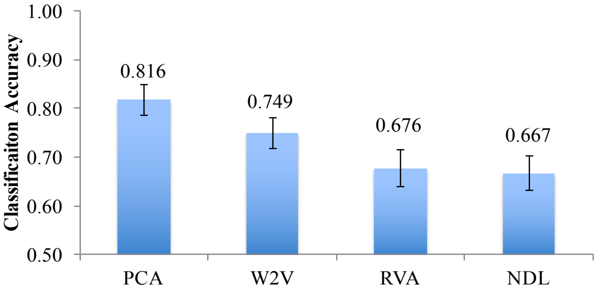

Figure 3. Mean classification accuracy (averaging across the 30 different categories and 95% CI) for the four DSMs.

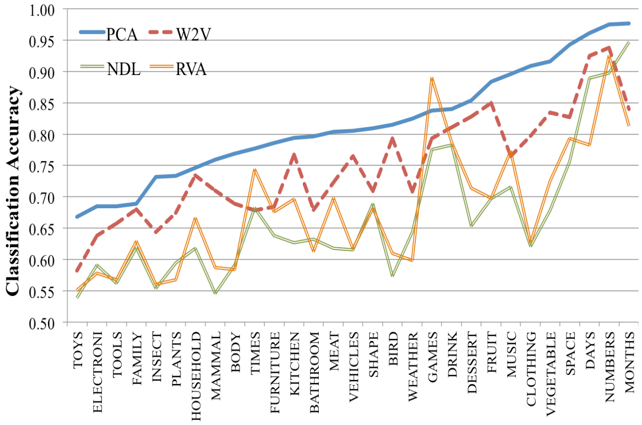

Figure 4. Mean classification accuracy on the 30 different categories for the four DSMs.

Finally, a comparison between the overall performances of the best co-occurrence model (PCA) and the best predictive mode (W2V) in this task revealed the superiority of the co-occurrence model. PCA includes a relatively sophisticated post-processing of the raw co-occurrence vectors, whereas W2V involves an error-based learning. Inputs to the both models were very similar, that is the co-occurrence information was collected from a window size of 12 from each side of a target word. As we mentioned above, the number of principal components in the PCA was a good equivalent to the size of the hidden layer in the W2V. If we compare it against the skipgram with 30 nodes in the hidden layer (in our previous experiment), we find an even bigger difference in the performance of the two models.

How models differ from one another qualitatively? Figure 4 provides an answer to this question by showing performance on each category separately. The models showed some variance in terms of what categories they learned better and what categories they learn worse. The less accurate models (NDL and RVA) tended to align fairly highly, doing relatively well and poorly on the same categories, and showing no significant difference in overall performance (t(29) = 1.48, p = 0.15). W2V outperformed both NDL and RVA, with a significantly higher overall performance (t(29) = 4.20, p < 0.001), and beating NDL on 28/30 categories and RVA on 27/30 categories. The PCA model performed the best, having a significantly higher overall performance than all three models (all *p*'s < .0001), and beating W2V on 30/30 categories, RVA on 29/30 categories, and NDL on 30/30 categories. Accuracy of a category did not correlate with its types or token frequency.

General Discussion

Our analyses compared two models that involved the abstraction of latent representations (PCA and W2V) versus two models that do not perform abstraction (RVA and NDL) when trained on child-directed speech. Both models that perform abstraction strongly outperformed those that do not. Our analyses also compared two co-occurrence models (PCA and RVA) with two predictive models (W2V and NDL). In previous work, W2V has been demonstrated to surpass standard co-occurrence models when trained on large amounts of data (e.g., Baroni et al., 2014; Mikolov, 2013). However, we found an interesting paradox: W2V was actually outperformed by a rather simple PCA-based co-occurrence model when applied to child-directed speech and with a classification of children's concepts. It is important to note that our PCA model was also in the list of models in Baroni et al. that were compared to W2V. Hence, the story isn't as simple as saying one model is "better" than the other. Despite the equivalence of the objective function optimization in W2V to the matrix factorization process of the PCA model (as pointed out in Levy & Goldberg 2014b), W2V involves an incremental error-driven learning process. It operates very well and efficiently with large amounts of data; also might be considered as a better simulation of the cognitive processes. On the other hand, our experiments show that PCA performs better on small and sparse linguistic data that children learn from.

Future work needs to focus on correlating the models' predictions with children's response data. In addition, our target words were all from the same syntactic category (nouns). Different distributional model architectures have been shown to vary in the type of information they learn best (e.g., syntactic vs. semantic association; Mikolov et al., 2013, or different types of semantic relations; Baroni et al., 2014; Jones, Kintsch, & Mewhort, 2006). Our classification task was one that required the models to learn paradigmatic similarity well. Interestingly, in the survey of Baroni et al. (2014), paradigmatic tasks were among the cases where W2V did not outperform the co-occurrence models. Hence, our results may point to the benefit of having complementary learning systems when constructing a semantic representation: An error-driven predictive mechanism and a mechanism applied to direct co-occurrences. There is no reason that the two learning mechanisms need to be mutually exclusive—there is a great deal of evidence that humans use both Hebbian and error-driven learning (e.g., O'Reilly, 1998). It is a reasonable presumption that perhaps the two systems work together to construct semantic memory from episodic experience.

Acknowledgement

This research was funded by grant R305A140382 from the Institute of Education Sciences.

References

Asr, F. T., Fazly, A. and Azimifar, Z. (2013). From cues to categories: a computational study of children's early word categorization. *Cognitive Aspects of Computational Language Acquisition*. (pp. 81-103).

Baayen, R. H., Shaoul, C., Willits, J. and Ramscar, M. (2015). Comprehension without segmentation: A proof of concept with naive discrimination learning. *Language, Cognition, and Neuroscience*.

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL* (pp. 238-247).

Bullinaria, J. A., & Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, *39*, 510-526.

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

Firth, J.R. (1957). A synopsis of linguistic theory 1930-1955. *In Studies in Linguistic Analysis*, (pp. 1-32).

Harris, Z. (1970). Distributional structure. In *Papers in Structural and Transformational Linguistics* (pp. 775–794).

Howard, M. W., Shankar, K. H., & Jagadisan, U. K. (2011). Constructing semantic representations from a gradually changing representation of temporal context. *Topics in Cognitive Science*, *3*, 48-73.

Jones, M. N., Kintsch, W., & Mewhort, D. J. K. (2006). High-dimensional semantic space accounts of priming. *Journal of Memory and Language*, 55, 534-552.

Jones, M. N., Willits, J., Dennis, S., & Jones, M. (2015). Models of semantic memory. *Oxford Handbook of Mathematical and Computational Psychology*, 232-254

Landauer, T. K. (2006). Latent semantic analysis. *Encyclopedia of Cognitive Science*.

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*(2), 211.

Levy, O. and Goldberg, Y., (2014a). Dependency-Based Word Embeddings. In *ACL (pp. 302-308).*

Levy, O., & Goldberg, Y. (2014b). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems (pp. 2177-2185).*

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, *28*(2), 203-208.

MacWhinney, B. (2000). *The CHILDES project: The database (Vol. 2).*

Miller, R. R., Barnet, R. C., and Grahame, N. J. (1995). Assessment of the rescorla-wagner model. *Psychological Bulletin*, 117(3). 363-386.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J. (2013). Efficient estimation of word representations in vector space. In *ICLR*.

O'Reilly, R. C. (1998). Six principles for biologically based computational models of cortical cognition. *Trends in cognitive sciences*, *2*, 455-462.

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*.

Rescorla, R. A. and Wagner, A. R. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In Black, A. H. and Prokasy, W. F., editors, *Classical conditioning II: Current research and theory*, pages 64-99.

Riordan, B., & Jones, M.N. (2011). Redundancy in perceptual and linguistic experience: Comparing feature-based and distributional models of semantic representation. *TopiCS*, *3*, 303-345.

Řehůřek, R., & Sojka, P. (2011). Gensim - Statistical Semantics in Python.

Rogers, T. T., & McClelland, J. L. (2004). *Semantic cognition: A parallel distributed processing approach*. MIT press.

Rohde, D. L., Gonnerman, L. M., & Plaut, D. C. (2006). An improved model of semantic similarity based on lexical co-occurrence. *Communications of the ACM*, *8*, 627-633.

Shaoul, C., Willits, J. A., Ramscar, M. Milin, P., & Baayen, R. H. (2016). A discrimination-driven model for the acquisition of lexical knowledge in auditory comprehension. *Manuscript under review*.

St. John, M., & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, *46*, 217-257.