Distributional Language Models and the Representation of Multiple Kinds of Semantic Relations

Jingfeng Zhang (jz44@illinois.edu)
Department of Psychology, 603 E Daniel St
Champaign, IL 61820 USA

Jon A. Willits (jwillits@illinois.edu)
Department of Psychology, 603 E Daniel St
Champaign, IL 61820 USA

Abstract

Distributional models (such as neural network language models) have been successfully used to model a wide range of linguistic semantic behaviors. However, they lack a way to distinctly represent different kinds of semantic relations within a single semantic space. Here, we propose that neural network language models can sensibly be interpreted as representing syntagmatic (co-occurrence) relations using their input-output mappings, and as representing paradigmatic (similarity) relations using the similarity of their internal representations. We tested and found support for this hypothesis on four neural network architectures (SRNs, LSTMs, Word2Vec and GPT-2) using a carefully constructed artificial language corpus. Using this corpus, we show that the models display interesting but understandable differences in their ability to represent these two kinds of relationships. This work demonstrates distributional models can simultaneously learn multiple kinds of relationships, and that systematic investigation of these models can lead to a deeper understanding of how they work.

Keywords: Computational modeling, Neural networks, Language learning, Statistical Learning, Semantic Memory

Background

The distributional hypothesis is a hypothesis for one way people learn semantic knowledge. From this perspective, representations for linguistic units include information about the contexts in which those units occur. Common to many distributional theories is that linguistic contexts can be used to learn two kinds of relationships: 1) paradigmatic relationship, the different categories to which a word belongs, and 2) syntagmatic relationships, the kinds of co-occurrence relationships a word can have (de Saussure, 1916/2011). There is considerable behavioral evidence that infants, children, and adults make use of distributional learning processes as a part of the language acquisition process (Unger & Fisher, 2021).

Distributional semantic models have undergone significant improvements in recent years. Distributional models have been used to semantically categorize words (Baroni & Lenci, 2010; Huebner & Willits, 2018; Riordan & Jones, 2011), model semantic priming (Mandera, Keuleers, & Brysbaert, 2017), and predict patterns of fMRI and EEG activation while processing language (Anderson, Kelley, & Maxwell, 2017; Michaelov & Bergen, 2022; Mitchell & Popham, 2008). More recently, models like GPT3 and ChatGPT showed significant success in applying distributional language models in various applied domains.

Although both computational and experimental work has shown that distributional information can be used to model many aspects of language and language processing, distributional models are still criticized for many reasons. One reason is that distributional models that represent information in a single semantic space (defined by the model's single, non-modular set of weights) cannot independently represent different kinds of relations (like syntagmatic and paradigmatic relationships)(Erk, 2016; Jones, Kintsch, & Mewhort, 2006; Mohammad, Kiritchenko, & Zhu, 2013). A second reason is that even when they do learn semantic relationships, they often require an extraordinary amount of data and training in order to do so. A final reason is that the models are so complex and difficult to understand, that it is often difficult to understand what it is they are and are not learning, and how they differ from other theories and models of representation.

In this paper, we conducted simulations to try to better understand different classes of neural networks models that learn distributional representations, and their strengths and weaknesses at learning certain kinds of relations (like syntagmatic and paradigmatic relations). We tested four successful neural network language models - the simple recurrent network (SRN), the Long-short term memory network (LSTM), the Word2Vec Skipgram model (W2V) and a GPT-like Transformer model on a carefully controlled artificial corpus to better understand these four models' differential capabilities at learning syntagmatic and paradigmatic relationships. A primary motivation of this work was to approach these models the way cognitive scientists treat theories and models of human psychology, trying to deeply understand how they work and what predictions the models make. If neural network models are going to be considered as candidate models of human cognition, we must have a much deeper understanding of how they work than we get from typical machine learning research.

Distributional Models of Relatedness

While the use of neural networks is currently very popular in machine learning and NLP, distributional language models originated as cognitive models. One of their main accomplishments is modeling relatedness in cognitive tasks. However, there are questions about the models' abilities to model different kinds of relationships. In earlier non-neural network models, some understanding existed regarding what kinds of models were better at different kinds of relationships. For example, Rubin, Kievit-Kylar, Willits, and

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.* ©2024 The Author(s). This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY).

Jones (2014) and Jones, Willits, and Dennis (2015) compared word-word co-occurrence models like HAL ((Lund & Burgess, 1996) and word-frequency-within-document models like LSA (Landauer & Dumais, 1997), showing that word-word co-occurrence models perform best at paradigmatic (similarity) relations, and word-document models perform best at syntagmatic (co-occurring) relations. However, no such understanding exists for the newer neural network-based language models. Neural networks like SRNs, LSTMs, Word2Vec and Transformer models are typically used as semantic models by using either computing the similarity of words' weight matrices or the similarity of the patterns of activation in their hidden layers when a word is activated. These models are quite good at modeling semantic similarity in general, but how do they fare on different kinds of relations?

In fact, there is an *a priori* reason to be skeptical that any single model *can* be good at modeling more than one kind of relation, at least in the way the models are currently used. Consider a hypothetical model whose nearest neighbors for *dog* are syntagmatic relations like *pet*, *leash* and *bark*. The better this model is at syntagmatic relations (the more of its nearest neighbors are these relations), but by definition, the fewer of its neighbors will be paradigmatic (similarity) relations like *cat*, *mouse*, and *wolf*. By definition, a model cannot do perfectly at both if it is using a single semantic space. As a model's nearest neighbors become more syntagmatic, this will crowd out paradigmatic relations, and vice versa.

Neural Network Models

Neural network models, however, potentially provide an account of the syntagmatic-paradigmatic distinction. Most neural network models are trained to predict word sequences, and use weighted connections and patterns of activation in hidden layers in order to do so. It has long been noted that there is a relationship between syntagmatic and paradigmatic relations that mirrors this distinction. Knowledge of paradigmatic categories or similarity may be the mechanism by which a person can make effective predictions about syntagmatic relationships (De Saussure et al., 1916; Elman, 1990; Lund & Burgess, 1996; Sloutsky, Yim, Yao, & Dennis, 2017).

Thus, it may make sense to think of neural network models not as one model but as two: 1) the pattern of activation on the output layer as a measure of a models' syntagmatic relations for an input word, and the pattern of activation at the hidden layer as a model of its paradigmatic structure. To the extent that the model's training objective is ordered word prediction, the output layer should be the better model of syntagmatic relations, and the hidden layer should be the better model of paradigmatic relations. In the following sections, we will describe how this general proposal leads to specific predictions for the four different models.

We proposed a predictive framework for different distributional models' performance of learning syntagmatic versus paradigmatic relations, positing that it is influenced by two primary factors. Firstly, the method of extracting a measure of relatedness from a model plays an important role. This extraction can be done in two approaches: 1) evaluating the output activation given an input, and 2) using the similarity of the "embeddings" of a model (such as its weight matrices or hidden state activations). We suggest that the former should more straightforwardly model words' syntagmatic relations, and the latter, as a measure of words' similarity or substitutability, should be a better model of words' paradigmatic relatedness. Secondly, the there are model-specific factors that should influence how good the models are at each of these kinds of relations. In this paper, we will examine four models (SRNs, LSTMs, GPT-style transformers, and Word2Vec), which have interesting differences corresponding to different theoretical claims about the nature of the cognitive or semantic system. One major difference is the extent to which the model has high encoding specificity for the exact word position. As we will describe, this should make a difference on the representations learned by the model.

SRNs. The SRN is an artificial neural network that activates one word at a time as an input. Activation is propagated through weighted connections to a hidden layer (that also receives recurrent input about its own state at the previous time step), which is then propagated through another set of weighted connections to an output layer. The output layer is trained using backpropagation to minimize error in activating the next word in the sequence (Elman, 1990). The SRN, by trying to predict the exact next word in a sequence, should have highly specific (and context-constrained) representations of co-occurrence relations. Words' embeddings (internal representations) should be similar to the extent that those words both predict the exact same set of next words in that exact context. One notable property of recurrent models is that the representations they learn are very constrained by the fact they are learning to predict in the forward direction. This means they are learning independent representations for each word based on what that word contributes to prediction that has not already been predicted by previous items in the sequence. If an SRN reliably sees a sequence like A-B-C, where A and B both predict C, it may actually only learn that A predicts B, and that A predicts C, but not that A predicts C (Huebner & Willits, 2023). And because it operates only in the forward direction, it won't learn a representation of C that has any relationship to the fact that it regularly comes after A, because knowing that fact is not useful in any way for predicting C comes after A. These constraints can really affect what internal representations it develops.

LSTMs. The Long Short-term Memory (LSTM) model is similar to an SRN, but with a more complex architecture in place of the SRN's recurrent hidden layer. The LSTM uses three multiplicative gating units to control the flow of information to and from a central unit (Hochreiter & Schmidhuber, 1997). The LSTM is even better than the SRN at its exact task, predicting the next word in that exact context. That means its output activations should be even better representations of syntagmatic relations. The LSTM's embeddings will

be even more context sensitive than the SRNs, as LSTMs are better at "remembering" the context of a word and using it for prediction. Thus, LSTMs embeddings will be even more context sensitive and specific, with earlier elements in a sequence encoding information about items coming downstream, and leading to that information not being encoded in the representations of the more adjacent words.

Transformers. Transformer models like GPT-2 (Radford et al., 2019) use a window of input words to predict the next word after the window. Transformers have a special kind of hidden layer called an attention layer that uses a unique method to determine which features should be attended to, depending on which words are in the window. The self-attention mechanism and the deepness of GPT-2 allows it to not only more accurate predict tokens in sequences but also generate high-dimensional linguistic space that represents words and sentences as vector projections (Hoover, Strobelt, & Gehrmann, 2019), (Ethayarajh, 2019). However, because GPT is not a recurrent model, but rather is operating over a fixed and specific window, this should change its behavior relative to SRNs and LSTMs. The ability to have perfect memory of the past set of words should improve GPT's ability to learn relations within its span, and completely eliminate its ability to learn relations outside that length. This will lead to GPT have an even more context specific representation of words in its embeddings. However, because the hidden layers of a transformer form a composite "gestalt" representation of the entire sentence at once and use that to predict the next word, that should lead to different outcomes compared to the recurrent models. In the service of learning the gestalt of the whole sequence, the GPT model might learn backward relationships, and might be more likely to distribute the co-ocurrence information more appropriately to the different words in a sequence, rather than encoding them in the first words in a sequence.

Word2Vec. Making predictions for Word2Vec Skipgram (W2V) is more complicated. W2V lacks the complex attnetion mechanisms of GPT. It is just a simple feedforward neural network with a single hidden layer. It activates a single word as its input, and tries to predict a randomly selected word from among the *n* words that come before or after that word in the sequence. In this way, W2V is very different from the other models because it is not trying to activate a specific word. This lack of encoding specificity may mean that the output activations of W2V might be more likely to include paradigmatic relations (substitutable words), and as a consequence its embeddings might be more likely to capture syntagmatic relationships.

Experiment

Artificial Corpus

In order to really understand the nature of how these neural networks operate, we opted to use a carefully constructed artificial corpus that systematically manipulated co-occurrence and similarity relationships. The corpus had a set of sequences that followed a simple four-token *AyB.* structure, similar to that used in previous studies of human and computational model-based statistical learning (Gómez & Maye, 2005; Willits, 2013). In each sentence, the word in position *A* belonged to a distributional category that allowed for perfect prediction of the category (and therefore the set of legal words) in position *B*. The corpus is designed to mimic natural language sentences like "dogs can eat", where the category of the first word (in this case, an animal) predicts the words that can occur in the third position(a verb that an animal can do). Each sentence in the corpus ended with a period.

In our corpus there were two *A* categories (*A1* and *A2*) and two *B* categories (*B1* and *B2*), each containing three words (e.g. *A1_1*, *A1_2*, and *A1_3* for category *A1*). The dependency between *A* and *B* was such that words in A1 had to be in a sentence with a word from *B1*, and words from *A2* had to be in a sentence with a word from *B2*. To continue with our natural language example, imagine that A1 words are animals (*dog*, *cat*, and *mouse*), and A2 words are vehicles (*car*, *truck*, and *bus*). B1 words are then verbs that can co-occur with animals but not vehicles (like *eat*, *drink*, and *watch*), and B2 words are words that can occur with vehicles but not animals (like *drive*, *park*, and *crash*).

Each of these *AB* pairs were intervened by a set of words in position *y*, which was equi-probable in our corpus with all combinations of *A* and *B* and thus created a long-distance dependency with no predictive value coming from the *y*-word. The corpus of all three word sequences used in the study is shown in Table 1.

Table 1: Stimulus inputs used in Study 1

A1_1	A1_2	A1_3
A1_1 y1 B1_2 .	A1_2 y1 B1_1 .	A1_3 y1 B1_1 .
A1_1 y2 B1_2 .	A1_2 y2 B1_1 .	A1_3 y2 B1_1 .
A1_1 y3 B1_2 .	A1_2 y3 B1_1 .	A1_3 y3 B1_1 .
A1_1 y1 B1_3 .	A1_2 y1 B1_3 .	A1_3 y1 B1_2 .
A1_1 y2 B1_3 .	A1_2 y2 B1_3 .	A1_3 y2 B1_2 .
A1_1 y3 B1_3 .	A1_2 y3 B1_3 .	A1_3 y3 B1_2 .
A2_1	A2_2	A2_3
A2_1 y1 B2_2 .	A2_2 y1 B2_1 .	A2_3 y1 B2_1 .
A2_1 y2 B2_2 .	A2_2 y2 B2_1 .	A2_3 y2 B2_1 .
A2_1 y3 B2_2 .	A2_2 y3 B2_1 .	A2_3 y3 B2_1 .
A2_1 y1 B2_3 .	A2_2 y1 B2_3 .	A2_3 y1 B2_2 .
A2_1 y2 B2_3 .	A2_2 y2 B2_3 .	A2_3 y2 B2_2 .
A2_1 y3 B2_3 .	A2_2 y3 B2_3 .	A2_3 y3 B2_2 .

This corpus will allow us to test each model's ability to learn paradigmatic structure, that all *A*, *y*, and *B*-words should be more similar to each other than to words from the other two categories; and that words should be more similar to the words from their subcategories (i.e. *A1*, *A2*, *B1*, and *B2*). We can also test each model's ability to learn syntagmatic structure, that *A*'s should precede *y*'s, that *y*'s should precede

B's, and critically, that the subcategories of *A* and *B* should be predictable from one another.

Model Training and Evaluation

For each of the four model types (SRN, LSTM, W2V, GPT) we trained 50 different randomly initialized models. Based on previous work using this size and complexity of an artificial corpus, each model had 16 hidden units. We trained each model for a number of epochs (2000) until its performance had reached asymptotic behavior.

Syntagmatic (Co-occurrence) Evaluation To evaluate each model's syntagmatic (co-occurrence) performance, we evaluated average output activation of the target words given the input word. Of particular interest was the the output activation given a y as input, in the context of a specific A-input. As an example, consider the sequence "A1_1, y1, B1_2 .". The y1 preceded by A1_1 has outputs that fall into the following categories: **period** (.); **y** (y1, y2, and y3); **present-B** (B1_2, the actual B that came next in the sentence); **legal-B** (B1_1 and B1_3, the B's from the legal category but which were not in that sentence); **illegal B** (B2_1, B2_2, and B2_3, the B's from the other category that couldn't co-occur with A1_1); **present-A** (A1_1, the output with the same label as the A in that sentence); **legal-A** (A1_2 and A1_3, the A's from the same subcategory as the A in that sentence); **illegal A** (A2_1, A2_2, and A2_3, the A's from the other subcategory that didn't occur in that sentence). Thus, if the model is correctly representing syntagmatic (co-occurring) relations, it should have only high activations for the present and legal B's, and low activations for the others. High values for present and legal A's would indicate the presence of paradigmatic (similarity) relationships, and high values for other units would indicate learning illegal or ungrammatical relationships.

Paradigmatic (Similarity) Evaluation. To evaluate each model's paradigmatic (similarity) performance, we evaluated the similarity of the models' weight vector from each input word to its first hidden layer. For example, in the SRN, the input unit "A1_1" had a set of weighted connections to its hidden layer, and a separate set of connections from "A1_2" to its hidden layer. We took these two weight vectors and computed the correlation to determine the similarity of A1_1 and A1_2. Similarity relations were computed over static weight matrices that did not vary by sentence, so the "legal" vs. "present" distinction was not present. Therefore, the comparisons of interest were just the similarity comparisons of A's to Legal A's, Illegal A's, Legal B's and Illegal B's. If the model's weights grouped an A with its Legal A's as most similar, followed by Illegal A's, and then followed by Legal B's and then Illegal B's, this would indicate a model with strong paradigmatic only relations. If instead the model's similarity space for an A had Legal B's more similar than Illegal B's, this would mean it would grouping items syntagmatically (by co-occurrence) rather than paradigmatically (by substitutability).

Results and Discussion

Output Activation as Syntagmatic Relation. For the prediction accuracy of the models, we compared the average output activation for categories of words that could occur in the next position, and compared that to the other categories. For the less interesting transitions in the sequence (.–¿A, A–¿y, B–¿.) the SRN, LSTM, and GPT models all reliably predicted the correct next element in the sequence, with other values near zero. W2V, because it predicts windows of words instead of the exact next word, had nonzero activations for other words, but still had the next items as the sequence among the highest output activation.

For the more critical comparison of what followed a y-word, we show the results in Figure 1. All four of the models correctly learned to predict a legal B after a y, a B that was allowed to co-occur with that A. None of the models were able to learn to predict the exact next B, as there was no statistical structure in the corpus giving them the information in order to do so. The sentences were presented in a random order, and since every legal B co-occurred with every A in its subcategory, there was no information that could be used to predict which exact B should occur. All models showed a very short period of overgeneralizing the B-prediction, and being equally happy with legal and illegal B's, before quickly learning to not activate the illegal Bs.

Thus, all four models' output activation served as a good model of syntagmatic (co-occurring) relations when rigorously defined and tested in this manner.

Weight Similarity as Paradigmatic Relation. Unlike their syntagmatic performance, the models varied considerably in the paradigmatic (similarity) performance. As predicted, SRNs had very high similarity scores for A's to Legal A's, and B's to Legal B's. Also as predicted, SRNs had very high similarity for B's to illegal B's. Given the forward directional nature of SRN's, they had little reason to distinguish B subcategories (Huebner & Willits, 2023). The SRN considered A's to be very dissimilar to illegal A's, as well considered all A's and B's to be very dissimilar to each other. The LSTM behaved qualitatively very similar behavior to the SRN, as one would expect since these models are very similar except for exactly how their recurrent hidden layers operate.

The GPT (Transformer) model also behaved very similar to the SRN and LSTM, with A's and B's very similar to their same subcategory members, and with A's very dissimilar from their opposite subcategory members, as should need to be the case to correctly predict which B comes after a particular A-y sequence. However, the transformer did show a bigger difference between B's and illegal B's than the SRN or LSTM, reflecting how Transformers, even though they don't need to represent different subcategories of B's, still end up doing so because of the way they represent the sequence as a whole.

Word2Vec, as expected, behaved quite different from the other three models. Word2Vec, because it was predicting the whole window rather than the next word, showed equally large differences between Legal and Illegal A's, and between

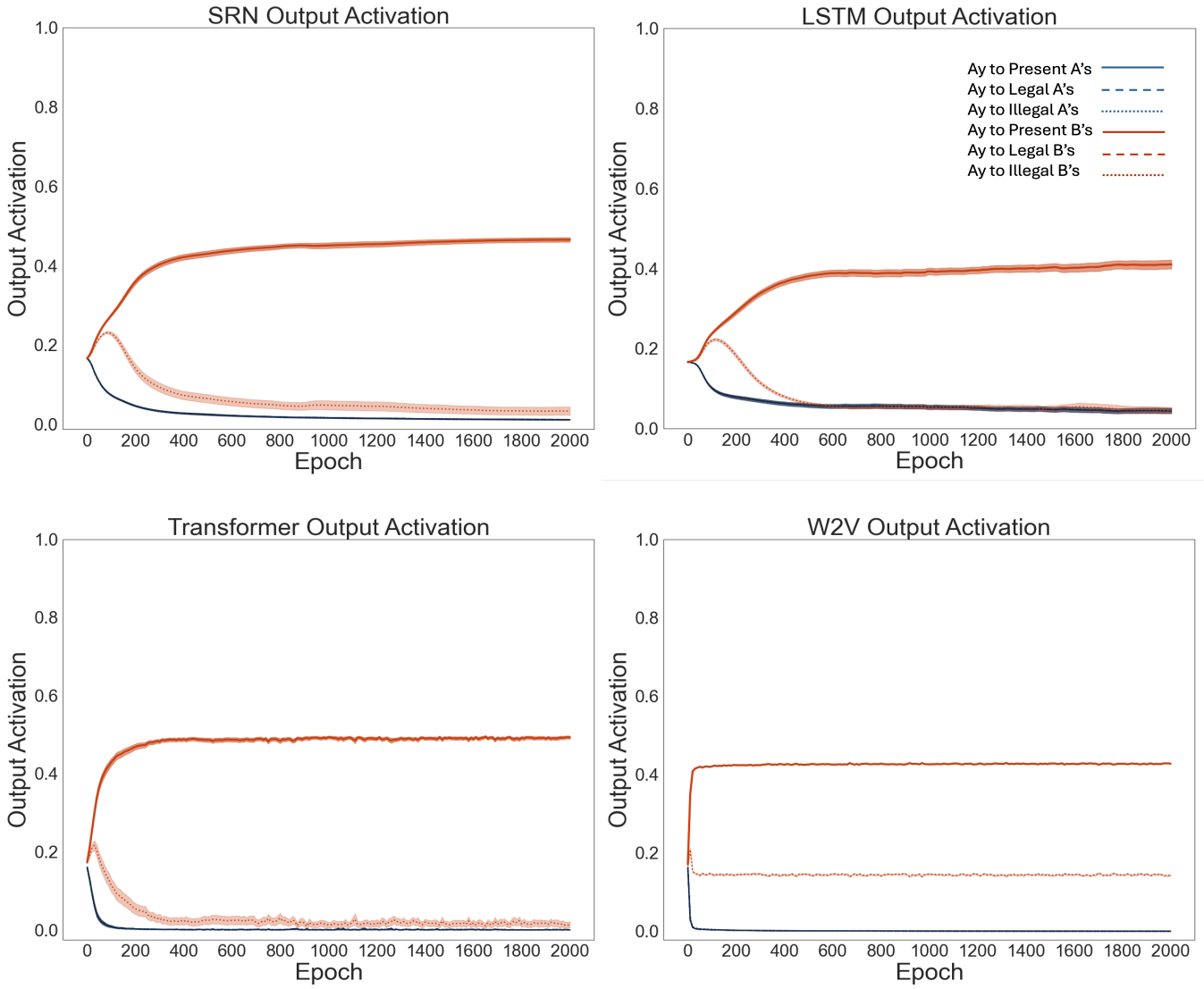

Figure 1: Mean output activation for all four models when input word is a y-word, in the context of a specific A-word. Solid orange lines denote activation of legal B-words, dotted orange line of illegal B-words. Blue lines (solid and dotted) represent the activation of legal and illegal A-words.

Legal and Illegal B's. Also unlike the other three models, for W2V the A-illegal A and B-Illegal B's were more similar to each other than A's were to Bs, or Bs were to As. Word2Vec, unlike any of the other models, seemed to more clearly be picking out the hierarchical nature of the category structure. This is interesting: the more a model was singularly focused on predicting the exact next word, the more it adjusted its similarity space to distinguish the words that were most likely to come next from all the other words. Word2Vec, which was trying to predict all the co-occurring words, developed a more complex similarity space.

General Discussion

In this research, we had one major goal, to use a carefully controlled artificial corpus to test neural network distributional language models, to see if they can be used to model both syntagmatic (co-occurring) and paradigmatic (similarity) relations. Our hypothesis was that this could be done by using the models' output layer activation as a model of syntagmatic relations, and the model's internal representational embeddings as a model of paradigmatic (similarity) relations. Our results are strongly in support of this hypothesis. This work has several important implications both for cognitive psychology and machine learning. The first implication pertains to the theoretical status of distributional learning mechanisms. This research adds additional evidence that distributional learning mechanisms may be a useful component of human language learning, with the clear and unambiguous demonstration that the models can distinctly learn and be used

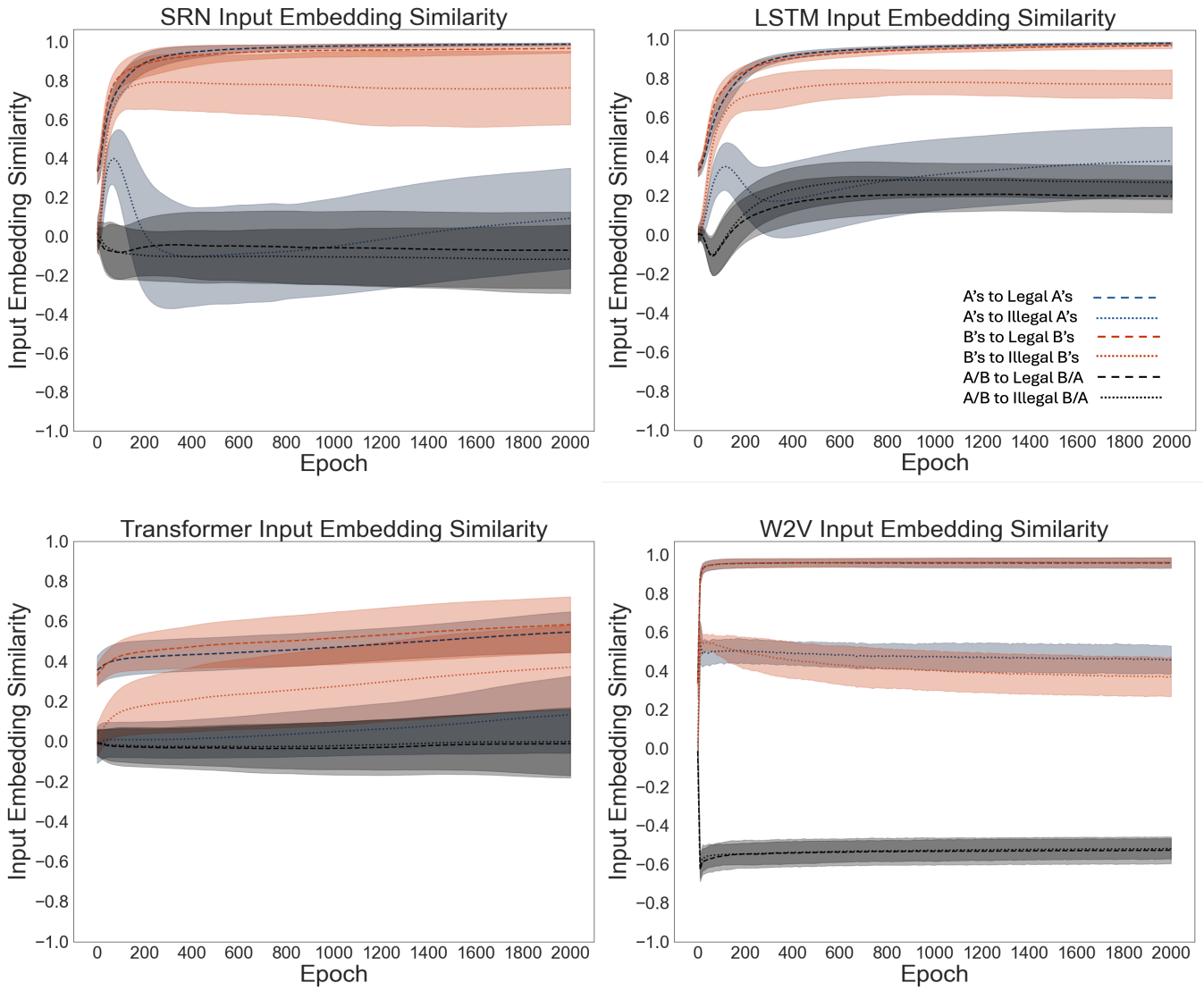

Figure 2: Mean similarity of input weight vectors for all four models. when input word is a y-word, in the context of a specific A-word. Dashed lines represent the similarity of words in the same subcategories (A in blue, B in orange, e.g., A1_1 with A1_2). Dotted lines represent the similarity of words in different subcategories (A in blue, B in orange, e.g., A1_1 with A2_2). Black lines represent the similarity of A words to B words: legal pairs dashed (e.g. A1_1-B1_1), illegal pairs dotted (e.g. A1_1-B2_1).

to model both kinds of relationships necessary for a linguistic system.

The second implication pertains to how these models are used (when trained on natural language corpora) to model cognitive psychology behavior experiments. This practice has become common, with considerable success (Kumar, 2021). However, one notable failure has been the ability to model different kinds of semantic relations simultaneously, and in particular a failure to model co-occurrence relations (Jones et al., 2006). With our research, the reason for this is now clear. All attempts to use these models have typically used exclusively the models' internal representations. But we have shown that in carefully controlled situations, these internal representations are actually quite bad at co-occurrence relations. Instead, researchers should use the model's output activations for model associations, and internal representations for modeling similarity. The third and final implication is about the proper treatment of complex neural network language, both as cognitive models but also as tools. This work clearly demonstrates that neural network models, despite their complexity, need not be treated as black boxes that cannot be understood. Careful experimental studies of how the models work can lead to a much deeper understanding of their principles, and allow us to better adjudicate both their plausibility as cognitive models, as well as the usefulness in applied situations.

References

Anderson, S. F., Kelley, K., & Maxwell, S. E. (2017). Sample-size planning for more accurate statistical power: A method adjusting sample effect sizes for publication bias and uncertainty. *Psychological science*, *28*(11), 1547–1562.

Baroni, M., & Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, *36*(4), 673–721.

de Saussure, F. (1916/2011). *Course in general linguistics*. Columbia University Press.

De Saussure, F., et al. (1916). Nature of the linguistic sign. *Course in general linguistics*, *1*, 65–70.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Erk, K. (2016). What do you know about an alligator when you know the company it keeps? *Semantics and Pragmatics*, *9*, 17–1.

Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*.

Gómez, R., & Maye, J. (2005). The developmental trajectory of nonadjacent dependency learning. *Infancy*, *7*(2), 183–206.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hoover, B., Strobelt, H., & Gehrmann, S. (2019). exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276*.

Huebner, P. A., & Willits, J. A. (2018). Structured semantic knowledge can emerge automatically from predicting word sequences in child-directed speech. *Frontiers in Psychology*, *9*, 133.

Huebner, P. A., & Willits, J. A. (2023). Analogical inference from distributional structure: What recurrent neural networks can tell us about word learning. *Machine Learning with Applications*, 100478.

Jones, M. N., Kintsch, W., & Mewhort, D. J. (2006). High-dimensional semantic space accounts of priming. *Journal of memory and language*, *55*(4), 534–552.

Jones, M. N., Willits, J., & Dennis, S. (2015). Models of semantic memory.

Kumar, A. A. (2021). Semantic memory: A review of methods, models, and current challenges. *Psychonomic Bulletin & Review*, *28*, 40–80.

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.

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, & computers*, *28*(2), 203–208.

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.

Michaelov, J. A., & Bergen, B. K. (2022). Collateral facilitation in humans and language models. *arXiv preprint arXiv:2211.05198*.

Mitchell, R., & Popham, F. (2008). Effect of exposure to natural environment on health inequalities: an observational population study. *The lancet*, *372*(9650), 1655–1660.

Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Riordan, B., & Jones, M. N. (2011). Redundancy in perceptual and linguistic experience: Comparing feature-based and distributional models of semantic representation. *Topics in Cognitive Science*, *3*(2), 303–345.

Rubin, T. N., Kievit-Kylar, B., Willits, J. A., & Jones, M. N. (2014). Organizing the space and behavior of semantic models. In *Cogsci... annual conference of the cognitive science society. cognitive science society (us). conference* (Vol. 2014, p. 1329).

Sloutsky, V. M., Yim, H., Yao, X., & Dennis, S. (2017). An associative account of the development of word learning. *Cognitive Psychology*, *97*, 1–30.

Unger, L., & Fisher, A. V. (2021). The emergence of richly organized semantic knowledge from simple statistics: A synthetic review. *Developmental Review*, *60*, 100949.

Willits, J. (2013). Learning nonadjacent dependencies in thought, language, and action: Not so hard after all. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 35).