Necessary Constraints on Continuous Distributional Semantic Models

Willa M. Mannering

Table of Contents

Question: *Review the literature on continuous learning DSMs, comparing and contrasting the various models, and include a discussion of emerging architectures in machine learning that are not necessarily models of human cognition but have shown success at semantic tasks and have promise at elucidating human mechanisms (e.g., feedforward and recurrent networks). What common trends emerge as necessary constraints on human mechanisms of continuous semantic learning?*

Introduction

Distributional models of semantic memory (DSMs; eg. Landauer & Dumais, 1997) have been hugely successful in cognitive science, explaining how humans transform first-order statistical experience with language into deep representations of word meaning. These models are all based on the distributional hypothesis from linguistics (Harris, 1954) and specify mechanisms to formalize the classic notion that "you shall know a word by the company it keeps" (Firth, 1957). The commonality to all DSMs is that they use co-occurrence counts of words across contexts in linguistic corpora and exploit these statistical redundancies to construct semantic representations. There are dozens of DSMs in the cognitive literature now, with learning mechanisms inspired by different theoretical camps ranging from Hebbian learning to probabilistic inference (for reviews, see Gunther, Rinaldi, & Marelli, 2019). In this paper I will consider multiple DSMs and stress the importance of sequential learning and contextualized representations as necessary constraints on human mechanisms of semantic learning. In the first section of this paper I will discuss a set of traditional DSMs to highlight the importance of sequential learning and implications each model has for human semantic memory and learning. Additionally, I will draw connections between the important mechanistic building blocks first introduced by these models and how they have informed modern DSMs. In the second section of this paper, I will discuss four modern DSMs to

highlight the importance of contextualized vectors. To do this I will introduce two static vector models and two contextualized vector models and discuss the advantages and disadvantages of both representation types while also discussing the implications each representation has for human semantic learning.

Sequential Learning in Traditional DSMs

In this section of the paper I will introduce three traditional DSMs: Latent Semantic Analysis (LSA), Simple Recurrent Networks (SRNs), and the Hyperspace Analogue to Language (HAL). Both HAL and the SRNs are able to learn sequentially while LSA cannot. Below I will delve into the advantages and disadvantages of both types of models and discuss implications for human semantic learning.

Latent Semantic Analysis (LSA)

One of the most popular DSM models within psychology is the Latent Semantic Analysis (LSA) model (Landauer & Dumais, 1997). To begin analysis, a word-by-document frequency matrix is created from a text corpus. This matrix has *n* rows, one for each word, and *m* columns, one for each document. Then, the raw frequency counts for each word are typically transformed in two ways. First, the frequencies are converted to its log and second, the log frequencies are weighted based on the entropy over documents within the corpus. The result of this transformation is that the relative importance of words which appear in many different contexts is reduced compared to words which only appear in few contexts. After this initial transformation, the matrix is compressed via Singular Value Decomposition (SVD). SVD is a form of factor analysis and allows any matrix to be decomposed into the product of three other matrices. After applying SVD to the matrix, the latent semantic components with the highest eigenvalues are kept. These

components exist in a reduced semantic space and represent how words co-occur over the analyzed corpus. Each word in the corpus is represented as a vector pattern over these latent semantic components. The result of this process is the creation of abstract representations of each word using the frequency of co-occurrence counts. The application of SVD to the word-by-document frequency matrix allows the second order relationships between words to emerge, so even if two words never directly occurred together they can have similar patterns over the latent semantic components. Typically, word vectors are compared by calculating the cosine similarity between the vectors.

LSA has successfully modeled many psychological phenomena and is still used within the field today. In the original paper, Landauer and Dumais (1997) showed that the model was able to fit human data in a variety of tasks including, the Test of English as a Foreign Language (TOEFL), semantic priming tasks, and the rate of acquisition of vocabulary knowledge of school children. Beyond the original model verifications, LSA is able to accurately model analogical reasoning (Ramscar & Yarlett, 2003) and predict text coherence and comprehension (Foltz, 1996). Additionally, LSA is still commonly used as a diagnostic tool for psychiatric and neurological populations. In fact, a recent meta-analysis of these types of studies determined that only 4 out of 21 studies used linguistic features from a model other than LSA (de Boer et al., 2018).

However, though LSA has had remarkable success in fitting human psychological data, one major objection to the model is the lack of cognitive plausibility. One criticism of LSA is that it is a bag-of-words model and makes no use of word-order information (Perfetti, 1998). This means that if LSA were to analyze a corpus and then analyze the same corpus where words were randomized within each document, the representations produced from both corpora would be exactly the same. These types of models are undesirable as word-order is an important source of information in language. Another, and possibly more serious criticism of LSA is that it is unable

to learn sequentially. In order to perform SVD on the word-by-document matrix, the entire corpus must be available. From a psychological standpoint, the creation of the word-by-document matrix can be said to represent a person's episodic memory, which is then abstracted via SVD into semantic memory. However, the inability of the model to learn sequentially suggests a form of learning which requires *all* episodic memories to be experienced *before* any semantic memories are created. Landauer and Dumais (1997) address this criticism by stating that they don't actually believe the brain performs SVD on a co-occurrence matrix but that it is some similar mathematical operation. This assertation, however, doesn't alter the psychological implications of the inability to learn sequentially.

Simple Recurrent Networks (SRN)

One of the earliest semantic models able to learn sequential dependencies in language were simple recurrent networks (SRN; Elman, 1990; Servan-Schreiber, Cleeremans, & McClelland, 1991). An SRN is a connectionist network that predicts input sequences by making use of a context layer. The context layer is created by copying the pattern of activation in the hidden layer onto the units in the context layer. The context units are then fed into the hidden layer again along with the input units. So, the context units act as a form of memory for the network and the SRN can remember sequential dependencies over multiple time steps. In these models, words are represented as the pattern of activation across the hidden layer. SRNs are able to predict the next word in a sequence (Elman, 1990) and can track some long-term dependencies (Elman, 1995). Unfortunately, these models were unable to scale up to natural language corpora and were limited in the amount of sequential information they could remember over many time steps.

While SRNs were not true DSMs, due to the limited amount of text data they could handle, the ideas introduced with these models were extremely important for the development of

continuous DSMs today. First, the notion of sequential dependency in language is a crucial component to any modern DSM and is critical to human language use. Second, the error-driven learning method employed by this model has been successfully used in modern predictive neural networks, an architecture which is applied to both semantic modeling and machine learning tasks. With modern computational advances, the predictive neural network is able to improve upon the fundamental aspects of the SRN and will be discussed in depth later in this paper.

Hyperspace Analogue to Language (HAL)

The Hyperspace Analogue to Language (HAL; Lund & Burgess, 1996) is one of the first true continuous DSMs and is able to handle much larger corpora than the SRN. This model uses a sliding window method to create vector representations of words within a corpus. The words in the window are recorded as having a higher co-occurrence strength the closer they are to each other in a sentence. For example, in the sentence, "The dog caught the ball", the word *dog* and *caught* would be coded as having a higher co-occurrence than *dog* and *ball*. The window is moved in one-word increments across the entire corpus. The result is a co-occurrence matrix with the entire corpus vocabulary as both axes, so each cell in the matrix represents the summed co-occurrence count of a word pair. The word pairs in HAL are direction sensitive, meaning the word pair AB may have a separate co-occurrence count than the word pair BA. The resulting semantic representation of each word is a vector of distance weighted co-occurrence values to all other words in the corpus. The distance between these word vectors is calculated by using the Minkowski distance metric,

$$Distance = \sqrt[r]{\sum(|x_i - y_i|)^r} \quad (1)$$

where r can be set to either 1 or 2. Typically, Euclidean distance (r = 2) is used for HAL but Lund and Burgess (1996) have shown that Manhattan distance (r = 1) is correlated with semantic priming in lexical decision as well. HAL has been successful in modeling a wide range of semantic and associative priming phenomena (Lund & Burgess, 1996) and introduced the popular sliding window method of computation.

One problem with HAL is that very frequent words, such as “a” or “and”, tend to cause similarity measures to be imbalanced. This is because most words will commonly co-occur with words such as “a” or “and” without conveying much about the semantic relatedness between the two words. There are a variety of methods to deal with this problem such as normalizing the co-occurrence matrix (e.g. COALS; Rohde, Gonnerman, & Plaut, 2006). While HAL is affected by this issue, it is not unique to co-occurrence models. This problem still exists in modern models and is commonly dealt with by deleting the 200 most common words from the corpus altogether during pre-processing. While this solution is not ideal, removing the most common words (often referred to as stop-words) allows semantic models to capture meaningful relationships between the remaining words in the corpus. Another problem with HAL is the size of the co-occurrence matrix. When applied to large natural language corpora, the co-occurrence matrix can get unmanageably large (Kanerva, 2009). Because of this, it is common in applications of HAL to select only a subset of columns within the co-occurrence matrix that have the most variance for further analysis (Lund & Burgess, 1996). While HAL is still able to successfully model semantic phenomena, this drawback raises some questions about the cognitive plausibility of constructing co-occurrence matrices. In response to the memory overflow issues, modern semantic models have largely abandoned the co-occurrence matrix in favor of vector representations.

Discussion

LSA, SRNs, and HAL each have different implications for semantic memory. LSA is able to efficiently leverage global statistical information from a text corpus and has been remarkably successful at fitting psychological data. However, LSA is unable to learn sequentially which not only raises questions of cognitive plausibility but also means it is unable to make use of local context data such as word order. SRNs were one of the first models to demonstrate the ability to learn sequential dependencies in language via predictive learning. While these models were unable to scale up to natural language corpora, the predictive learning method has been very successful in modern predictive neural networks which dominated the NLP field for a number of years. Finally, HAL introduced the sliding window method which has been used by many successive models, such as predictive neural networks and random vector accumulation models. This method allows HAL to learn sequentially and make use of local word context. However, HAL has problems with memory overflow due to the construction of a co-occurrence matrix. While this is not a problem when using smaller corpora, modern text corpora can contain billions of words—necessitating a more compact way to create and store word representations.

Representation in Modern DSMs

The widely accepted solution to memory overflow caused by co-occurrence matrices is the adoption of the vector word representation. There are two primary types of vector representations: static and contextualized. In the following section I will introduce and discuss four modern DSMs: two that use static representations and two that use contextualized representations. One important topic that I will discuss in this section is word sense disambiguation—a task given to semantic models to correctly identify the appropriate sense of a word given context. Words with multiple senses are often very difficult for DSMs to handle correctly and is one major differentiating factor

between static and contextualized representations. Consequently, the ability to correctly represent words with multiple senses is a constraint on DSMs I believe should necessarily be included in any new state-of-the-art models.

Static Representations

Models which produce static representations have the most difficulty with word sense disambiguation because there is only one, averaged representation for each vocabulary word. Thus, if a word has multiple senses, multiple contexts for a single word ends up averaged into the final representation of that word—meaning the final representation ends up "split" in semantic space between each word sense. While all models that create static representations struggle with word sense disambiguation, some models have serious practical issues when it comes to learning vector representations of words with multiple senses in a sequential manner. The two static representation models I will be comparing are predictive neural networks, which are known to catastrophically forget word senses when learning sequentially, and random vector accumulation models which do not have problems with catastrophic forgetting. In this section I will outline the basic learning mechanism of each model, discuss catastrophic forgetting and why it occurs in predictive neural networks, and discuss the implications each model has for human semantic learning and memory.

Predictive Neural Networks

Recently, there has been a resurgence of neural network models across cognitive science and machine learning. This resurgence has included very successful predictive DSMs within connectionist architectures based on principles of error-driven learning derived from theories of reinforcement learning. Earlier work explored neural networks to learn distributed semantic representations from artificial languages using simple recurrent networks (e.g., Elman, 1990) and

feed-forward networks (Rogers & McClelland, 2004). Neural networks were essentially non-existent in the 1990s and early 2000s as they couldn't scale up and learn from natural language corpora. Rather, the field became fixated on algebraic models based on dimensional reduction mechanisms such as LSA (Landauer & Dumais, 1997).

The standard predictive network currently discussed in the literature is the word2vec model (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). Word2vec is a feedforward neural network with localist input and output layers that contain one node per word in the vocabulary, and a hidden layer of ~300 nodes that is fully connected to both input and output layers. The word2vec architecture has two possible model directions: The context may be used to predict the word (referred to as a CBOW), or the word may be used to predict the context (a skipgram). The skipgram model direction will be used as the demonstration example because it maps conceptually onto most connectionist models of cognition. When a linguistic context is sampled (e.g., "save money bank") the target node is activated at the input layer (+bank) and activation is forward propagated to the output layer, with the desired output being the observed context words (+save, +money). The error signal (observed output – desired output) is applied to the network with backpropagation (Rumelhart, Hinton, & Williams, 1986) to correct the weights and make it more likely that the correct output pattern will be generated the next time this target word is encountered. The semantic representations are created by exporting the final pattern of weights across the input-to-hidden layer. Two words that have similar vector patterns across these weights are predicted by similar contexts, even if they never co-occur with each other, akin to (but superior in data fit) the second-order inference vectors learned by traditional DSMs.

Word2vec has received considerable attention in the machine learning literature due to its ability to outperform all previous DSMs (Baroni, Dinu, & Kruszewski, 2014). To cognitive science, this success is of considerable interest as word2vec implements a potentially biologically

plausible neural architecture and links to classic theories of reinforcement learning (e.g., Rescorla & Wagner, 1972), drawing theoretical connections to other areas of cognition with a unified mechanism. One feature of particular interest in these models is that they are sequential learners, in contrast to earlier algebraic DSMs, like LSA, which were unable to learn sequentially. The sequential learning of predictive neural networks has been taken by some as additional evidence in favor of cognitive plausibility of the models. Additionally, Mandera, Keuleers, and Brysbaert (2017) argue that a predictive model is a theoretical leap forward as they solve the memory overflow issues co-occurrence models such as HAL faced by not constructing a co-occurrence matrix at all.

While the hype surrounding predictive neural networks is warranted given their success at fitting human data, it is important to remember that the models also inherit the weaknesses of their predecessors. One weakness in particular that models such as word2vec are likely to exhibit is *catastrophic forgetting (CF):* The tendency of neural networks to completely loose previously learned associations when encoding new ones. In McCloskey and Cohen's (1989) seminal work on catastrophic forgetting, a standard neural network was trained to learn single-digit "ones" arithmetic facts (e.g., 1 + 1, 9 + 1) using backpropagation until the network had perfectly learned the associations. They next trained the same network on a new set of single-digit "twos" facts (e.g., 2 + 1), until the network had been trained to respond correctly to all of them. While the network was able to correctly answer the twos facts, it had completely lost the previously learned ones facts—the associations that had been trained to zero error were now lost completely. The learning of new associations with backpropagation overwrote the previous learning. The CF pattern was duplicated in a second experiment by McCloskey and Cohen by simulating standard tasks of paired associate word learning. Further, Ratcliff (1990) demonstrated that in standard sequential learning

paradigms, backpropagation networks catastrophically forget previous items as new items are sequentially learned, unlike humans performing the same tasks.

Catastrophic forgetting in word2vec occurs when the model attempts to learn words with multiple senses in a sequential fashion (Mannering & Jones, 2020). For example, take the word *bank* which has at least two senses: money-bank and river-bank. Due to CF, if the model learns the money sense before learning the river sense, then the money sense will be forgotten—skewing the semantic representation of *bank* towards the river sense of the word. Figure 1 illustrates this situation. The left panel shows the ideal situation if the contexts of a corpus were randomly sampled, with *bank* equidistant to its two senses. The right panel shows the case where river contexts have been sampled most recently—in this case, *bank* is biased by backpropagation to the more recent sense despite the fact that it is equally frequent in the corpus as the money sense.

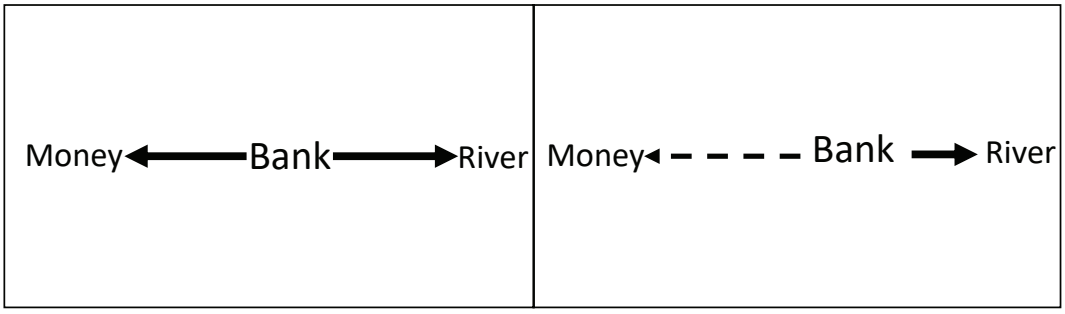

Figure 1. A schematic illustration of the semantic space of *bank* as a function of sense learning order

Additionally, while the backpropagation algorithm links to classic theories of reinforcement learning, it has been criticized for being biologically implausible. Some popular criticisms are, among others, (1) that the computation is only linear, whereas biological systems contain both linear and non-linear processes, (2) that biological neurons communicate via binary values not continuous values, and (3) if backpropagation were used in the brain, the feedback paths would need to use precisely symmetric weights to the feedforward paths (known as the Weight

Transport Problem: Benigo, Lee, Bornschein, Mesnard, & Lin, 2015; Grossberg, 1987). While there are many concerns about the biological plausibility of backpropagation *as it is now* there is a huge ongoing effort within the neuroscience and machine learning fields to alter the algorithm to make it more plausible.

Preventing Catastrophic Forgetting in Predictive DSMs

Recently, algorithms that alter the plasticity of weights and weight update functions of neural networks have become the de facto solutions for catastrophic forgetting within the machine learning field. One such algorithm is Elastic Weight Consolidation (EWC) which has shown great promise at learning new associations while insulating previously learned associations against forgetting (Kirkpatrick et al., 2017). EWC constrains the weight space of a deep learning network within the optimal parameter space of a previously learned task, essentially having the effect of a spring (or elastic) on already learned weights that are important to a previously learned task. Kirkpatrick et al. demonstrated that EWC networks could learn new strategies and associative patterns with minimal loss to previously learned but orthogonal associative patterns. However, Mannering and Jones (2020) found that EWC was unable to prevent catastrophic forgetting when applied to DSMs. They argue that EWC as it is currently implemented is not theoretically plausible for any task which requires unsupervised learning because the new loss function must be "switched on" when the network is learning a second task, i.e. the network needs to be externally told when senses are changing, which skirts the problem of learning the signal. This is especially cumbersome in NLP where it is impossible to supervise learning to the extent which EWC requires. Furthermore, EWC is unable to scale up well with its current implementation. These results suggest that efforts to mitigate the effects of catastrophic forgetting need to be interdisciplinary. Within the machine learning and the neuroscience community, insulating, or "vaccinating",

predictive neural networks from catastrophic forgetting is an emerging area that has seen some innovation in recent years. However, these types of solutions only consider the problem as it relates to strictly machine learning tasks such as categorization or image classification tasks.

Other possible solutions from the machine learning field include dropout (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012) and Gradient Episodic Memory (Lopez-Paz & Ranzato, 2017). Dropout is an addition to stochastic gradient descent training in which the input and hidden layers of a network are multiplied by a binary mask with each training instance that is learned. In effect, this allows many different networks to be trained on subsets of the training set and the resulting model predictions to be averaged together, effectively regularizing the model's predictions over sequenced training sets. Although it has been claimed that dropout acts somewhat similar to how hippocampal-neocortical complimentary encoding systems function (Goodfellow, Mirza, Xiao, Courville, & Bengio, 2013) the link is rather tenuous, and is the same theoretical claim that is made with EWC and various other algorithms that are computationally quite different. Other practical approaches include Gradient Episodic Memory (Lopez-Paz & Ranzato, 2017), which basically retains exemplars of previous tasks when learning new ones. But none of these approaches are likely to be adaptable to the learning of multiple senses of words, and all lack cognitive and neural plausibility (for a review see Parisi, Kemker, Part, Kanan, & Wertmer, 2019). However, while these approaches have seen some success in mitigating catastrophic forgetting when applied to machine learning tasks, there is some doubt as to whether these solutions would be adequate when used with a DSM. Often times, these types of solutions are not theoretically or practically feasible in the field of semantic modeling due to the huge amount of data necessary to train DSMs.

Other possible solutions take their organizations from the anatomical and functional organization of the brain. For example, complimentary systems theory (McClelland, McNaughton,

& O'Reilly, 1995) posits that a slow neocortical processing system and faster hippocampal encoding system in the human brain allows deep processing of predictive information while avoiding problems that come with CF. The neocortical system learns deep abstractions but is susceptible to CF, however, it is complimented by the hippocampal replay of recent episodic experience. More recent neurobiological models of memory, such as the cascade model (Benna & Fusi, 2015), manage to avoid problems of CF by implementing a bimodal system that initially learns very rapidly but gradually transfers information to a slower-learning mechanism. For reinforcement learning of procedural tasks in machine learning, the most successful convolution networks of visual perception use "experiential playback" to allow them to randomize over the training input that is otherwise experienced in a serial manner (Mnih et al., 2015), very much like complimentary learning systems theory.

Finally, other architectures, which I will discuss later in this paper, such as exemplar-based models and random vector accumulation models, could also be solutions to CF. These architectures are theoretically immune to catastrophic forgetting and incorporate different theoretical frameworks of learning (see M.N. Jones, Willits, & Dennis, 2015). Exemplar-based models, unlike other DSM models which store an abstract semantic representation, store only episodic context (M. N. Jones, 2018). These models construct semantic meaning from the aggregation of episodic context when presented with a memory cue (Jamieson, Johns, Avery, & Jones, 2018) and produce contextualized representations. Random vector accumulation models, which I will discuss next, should be immune to catastrophic forgetting because they utilize principles of associative learning and do not rely on an error signal—learning via a simple Hebbian co-occurrence learning mechanism.

Random Vector Accumulation (RVA) Models

Unlike predictive neural networks, which are affected by CF due to the error signal produced during learning, RVA models have been shown to be immune to CF (Mannering & Jones, 2020) most likely because they utilize principles of associative learning and do not rely on an error signal to learn. These models learn via a simple Hebbian co-occurrence learning mechanism. A prominent example of an RVA model is the Bound Encoding of the Aggregate Language Environment (BEAGLE; M. N. Jones & Mewhort, 2007) model. Which is, at its core an RVA with the additional ability to learn and represent order information. The most basic RVAs first begin by initializing two random vectors from an arbitrary distribution and of arbitrary dimensionality for each word encountered in a corpus. One of these vectors, called the environment vector, is unique to each word in the vocabulary, and the other, the memory vector, is a summation of all context words. The update function for the memory vector of each word in the vocabulary is described in Equation 2:

$$m_i = e_{i-1} + e_{i+1} \tag{2}$$

where m_i is the memory vector for an arbitrary word in a corpus, e_{i-1} is the unique environment vector for the context word before *i,* and e_{i+1} is the unique environment vector for the context vector after *i.* So, the memory vector for word *i* stores the context vectors for every other word that appears in context with word *i.*

In addition to the memory vectors created for each word, BEAGLE calculates an order vector via circular convolution for words in a given sentence. The order information for a word, w, is produce by binding it to all n-gram chunks in the same sentence as w. The position of w is represented via a placeholder vector, Φ, which is then convolved with the environmental vectors of the words surrounding w in the sentence. Similar to the environmental vectors, Φ is held constant across training. The bindings created for w are then summed together to produce the order

vector specific to w. Take for example the sentence, “A happy cat”. The bindings for the word “happy” would consist of two bigrams and one trigram:

$$bind_{happy,1} = e_A * \Phi$$

$$bind_{happy,2} = \Phi * e_{cat}$$

$$bind_{happy,3} = e_A * \Phi * e_{cat} \quad (3)$$

and the order vector for the word “happy” is calculated by:

$$\boldsymbol{o}_{happy} = \sum_{j=1}^{n=3} bind_{happy,j} \quad (4)$$

where *n* is the total number of convolution bindings produced for the word “happy” and $bind_{happy,j}$ represents the j^{th} binding for “happy”.

BEAGLE has been shown to successfully model numerous semantic phenomena such as semantic priming and fitting semantic distance norms (M. N. Jones & Mewhort, 2007). BEAGLE improves on previous DSMs such as LSA by incorporating word order information and by implementing a sequential learning algorithm. BEAGLE, and other RVA models, are also able to learn sequentially without incurring catastrophic forgetting like predictive neural networks due to their lack of error-signal.

However, while RVA models do not face difficulties with sequential learning like predictive neural networks do, they have faced criticisms in the past. RVA models in particular are known to have problems with metric space compression—causing most word similarities to be compressed between 0 and 1—which limits the ability of the model to discriminate between related and unrelated words (Asr & Jones, 2017). It was initially believed that predictive neural networks were able to more accurately discriminate between words because of back-propagation or the connectionist architectures they commonly use (which is one of the reasons this architecture has become so popular). However, recently the role of negative sampling in DSMs has been explored

in more depth by Johns, Mewhort, and Jones (2019) who find that the success predictive neural networks have at discriminating between words is due to the inclusion of negative information in the training data—not the use of connectionist architecture or predictive error correction. In fact, Johns et al. demonstrated when negative sampling information is included in the training data for other DSMs, including RVA models, their ability to discriminate words is on par with predictive neural networks. This indicates that it is not error correction that is producing the benefit of predictive models, but the benefit can be seen in errorless Hebbian learning models if they also implement negative sampling.

While RVA models may seem especially suited to replace predictive neural networks given their recent ability to incorporate negative sampling information, they still share the same pitfalls as word2vec when it comes to their static representations. RVA models, as well as word2vec, produce a single, averaged word vector per word in the vocabulary, meaning they also have trouble with word sense disambiguation. So, while the RVA model will not catastrophically forget one sense of a word when learning a new sense, the representation will be "split" in semantic space between the word's various meanings. Unfortunately, this problem is universal to all models which produce static vector representations.

Contextualized Representations

In the next section of the paper, I will introduce and discuss DSMs which produce contextualized representations. A recurring problem with the previously discussed models is the static representations. While some models like the RVA do not catastrophically forget word senses when learning sequentially, they are still unable to correctly represent words with multiple senses. One architecture that I will discuss is an exemplar-based DSM which is able to store all context information in memory and adjust a word's representation based on context at the time of retrieval.

After this I will explain and discuss the Transformer and BERT models which have recently taken the NLP field by storm. These models are able to create contextualized word representations while avoiding some of the downfalls of the exemplar-based DSMs and have been hugely successful at modeling large natural language corpora. However, having not originated from a cognitive field like the exemplar-based DSM, it is less clear what the implications for human learning and semantic memory are for these models.

Exemplar-Based DSMs

So far, the models discussed in this paper have all produced static word representations. That is, they create a single, averaged vector representation for each word in the corpus vocabulary. However, the static representations used by most modern DSMs, akin to the prototype model from the category learning field, are at odds with current directions in the fields of categorization and episodic memory. A classic debate within the field of category learning is between the prototype and exemplar-based models. Exemplar models suggest that people represent categories by storing individual memory traces (Nosofsky, 1986) while prototype models suggest that people create a "prototypical" representation of a category, which is typically an average computed over the training items (Reed, 1972). Within the categorization literature, the prototype model is almost unanimously considered inferior to the exemplar-based models because prototype models are unable to account for human behavior when category structures are nonlinear. Additionally, exemplar models have been shown to better predict human behavior even when the category structures are linear (Stanton, Nosofsky, & Zaki, 2002). M. N. Jones (2017) suggests that the prototype-style DSMs that produce static representations experience the same disadvantages as the prototype models of categorization.

One solution to the problems of static representations is to use an exemplar-based DSM which produces contextualized representations. These models, similar to the exemplar models in the categorization literature, represent memory as a word-by-context matrix. That is, every time a word appears in a corpus the surrounding context is recorded. Then, when the semantic representation is retrieved it is constructed on the fly as the average of other words in memory, weighted by their similarity to the target word. Thus, these models represent a different theoretical approach to semantic abstraction. The difference between most prototype DSMs is the learning mechanism for creating the final static representations. Prototype DSMs posit that semantic abstraction occurs at encoding, thus the learning mechanism each model employs is typically the point of interest. The exemplar-based DSM, however, posits that semantic abstraction does not come about via a learning mechanism but is a byproduct of retrieval. Jamieson et al. (2018) have successfully implemented an exemplar-based DSM called the Instance Theory of Semantics (ITS) model. ITS was not only able to reproduce classic semantic phenomena commonly used to support prototype DSMs, but was also able to correctly represent words with multiple senses where prototype models struggle due to their static semantic representations.

In addition to being able to correctly handle represent words with multiple senses, exemplar-based DSMs should also theoretically immune to CF when learning sequentially. These models do not use back-propagation, which is commonly thought to be the underlying cause of CF. However, while these models could be a promising solution to CF and word sense disambiguation, they have one major drawback. These models are designed to store the context information every time a vocabulary word appears in the corpus. While this mechanism gives these models an advantage over prototype DSMs when it comes to word sense disambiguation, it also limits the corpus size these models can handle. There are only so many context instances that can be stored in memory before space begins to run out. This problem is similar to what HAL faced

and unfortunately, there is no agreed-upon solution, putting these models at a disadvantage when it comes to popular NLP applications of DSMs.

Transformers

The Transformer architecture (Vaswani et al., 2017) originated in the machine learning field and is the basis for many current state-of-the-art machine translation and language models (such as BERT). Because the Transformer architecture originated in the machine learning field, it has a different history than the DSMs I've previously discussed in this paper. Models developed in the machine learning or NLP fields are often used for different purposes than DSMs. While some models can be considered semantic models, often times the people using them are more interested in practicality than theoretical implications of the potential cognitive mechanisms involved. While the Transformer architecture originated from the machine learning field and has mostly been applied to NLP rather than cognitive tasks, it is useful to learn about because it is able to create contextualized representations, performs extremely well on tasks all DSMs are expected to perform, learns sequentially, and introduces an attention mechanism that could prove to be an interesting addition to the cognitive mechanisms used by DSMs.

Because the Transformer originated in the machine learning field, it is more similar to various older NLP architectures, such as the Long-Short-Term-Memory (LSTM) models, than older DSMs. The key innovation of the Transformer is that it utilizes a new attention mechanism without using a Recurrent Neural Network (RNN). Until recently, RNNs were one of the most common ways to capture sequential dependencies in models used for NLP tasks. Unfortunately, RNNs are slow and have issues learning long term relationships between words due to the vanishing gradient problem—a problem with gradient based learning functions that causes networks with many layers to be difficult to train. Vaswani et al. (2017) were able to show that the

Transformer, an architecture with *only* the attention mechanisms and no RNN, was able to outperform other sequential NLP models, such as the LSTM and RNN, in translation and other NLP tasks while also being able to more efficiently train on large data sets due to the parallel nature of the input data.

The Transformer architecture can be seen below in Figure 2.

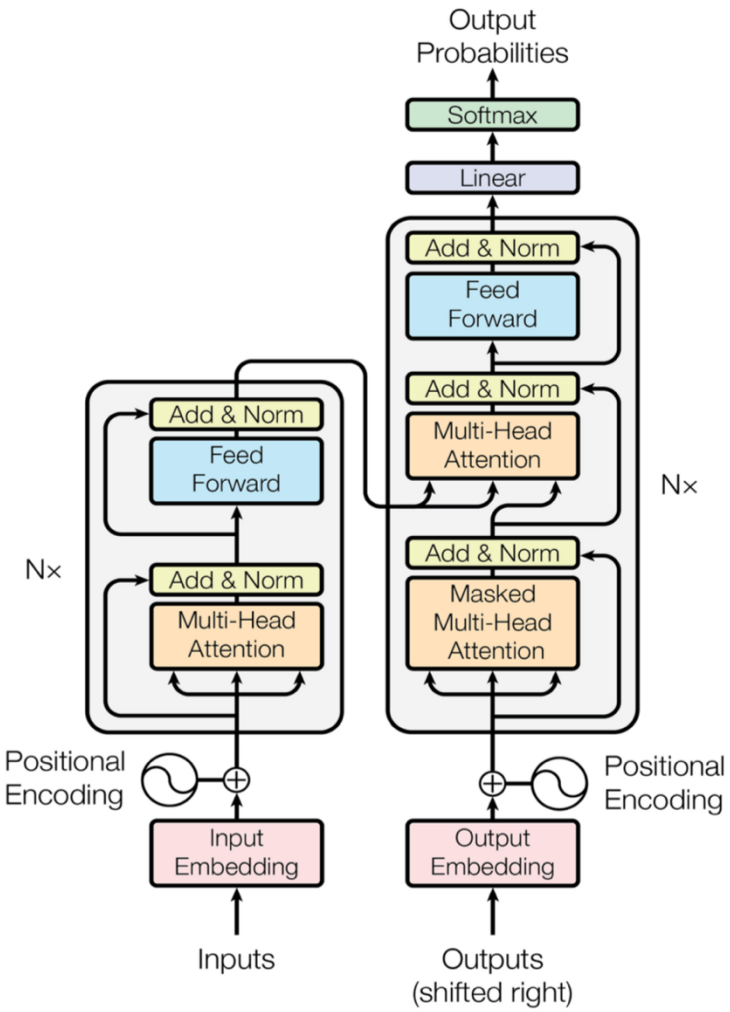

Figure 2. Transformer architecture from 'Attention Is All You Need' by Vaswani et al. (2017)

The module group on the left makes up the Encoder and on the module group on the right makes up the Decoder. The Encoder takes the original sequence of input words and maps it to a high dimensional space. Then, the Decoder takes this mapping and translates it into a new sequence, typically a different language. Both the Encoder and Decoder consist of modules which are stacked on top of each other multiple times and commonly consist of a Multi-Head Attention layer and a Feed Forward Network layer. The basic Encoder and Decoder architecture used in the Transformer model is not a new feature and has been used before in LSTM models. Attention mechanisms have been introduced and used before the Transformer architecture as well. In fact, the first attention mechanisms were added as an additional feature to the RNN architecture (Bahdanau, Cho, & Bengio, 2014) allowing them to remember longer sequences of input. Vaswani et al. (2017) found that the attention mechanism was able to capture sequential dependencies without the RNN architecture, greatly reducing training time. Because the attention mechanism receives input in chunks, unlike RNNs which receive input one-by-one, they can learn the dependencies between inputs all at once and are highly parallelizable—a desirable feature in large-scale NLP models. While the attention mechanism itself isn't new, the use of Multi-Head Attention layers is the truly innovative aspect of Transformers.

The attention mechanism is responsible for the contextualized representations produced by Transformers. To begin, the attention mechanism must be given context-free vector representations as inputs. These representations are typically produced by an embedding layer and are similar to the embeddings produced by models like word2vec. These initial vectors are static and represent the average meaning of a word regardless of context. The goal of the attention mechanism after receiving these static representations, is to learn to dynamically transform the representations based on context (Note that the attention mechanism receives input in chunks, thus it is able to make use of the context provided by entire sentences). This happens in three steps.

First, for each word in a sentence, w_i, the model must calculate how similar w_i is to every other word in the sentence. Vaswani et al. (2017) calculate the similarity of each word in a sentence as:

$$Q \cdot Q^T \tag{5}$$

where Q is a matrix of shape (input length, embedding dimensions). The resulting matrix is of shape (input length, input length) where $Q_{i,j}$ gives the similarity between words w_i and w_j. Second, once the similarities have been calculated, the attention mechanism must calculate attention scores of w_i towards all other words in the sentence. This is done by taking the softmax of the similarity matrix by row:

$$softmax\left(\frac{Q \cdot Q^T}{\sqrt{embedding\_dimensions}}\right) \tag{6}$$

The result is a matrix with values between 0 and 1 whose rows each sum up to 1. For each word in the sentence, w_i, attention scores are higher for words that are very similar to w_i. Based on the context of the input sentence, different surrounding words can be given varying amounts of attention. For example, take the sentences "I am eating bass" and "I am eating undercooked bass" where w_i = bass. In the first sentence, eating is the only word that's very similar to bass, thus the attention of bass will be fully directed towards eating. However, in the second sentence, both eating and undercooked are similar to bass so the attention of bass will be split between the two words. In this way, attention must be distributed among all related words in a sentence.

The third and final step to transforming word representations based on context is to update the static representations provided to the attention mechanism as input. To do this, the resulting matrix from step 2 (Equation 5) is multiplied with matrix Q. The result for each word in the sentence, w_i, is a new representation that is the weighted average of every other word in the sentence with the weights given by the calculated attention scores. Thus, if the input sentence were,

“I like to eat bass”, the final representation of the word bass would be shifted towards the fish-sense of the word whereas if the input sentence were, “I like playing the bass”, the final representation of the word bass would be shifted towards the guitar-sense of the word.

Given only one attention layer, it is not obvious how the Transformer architecture should be able to produce contextualized representations that handle words with multiple senses. Though the vectors change based on the input context, the output of the individual attention layers are still single vector representations per word. Meaning, if one attention layer encountered a word in multiple contexts, the averaged representation would get shifted around to match the most recent context. However, this potential problem is avoided via the Multi-Head Attention mechanism implemented in the Transformer framework. The Multi-Head Attention mechanism allows multiple word representations to be calculated in parallel, producing multiple sets of attention scores. Each individual attention layer is theoretically supposed to capture a particular linguistic property of the input. Then, the final contextualized word representations are created by concatenating each of the representations calculated by the individual attention layers—resulting in a contextualized vector representation that can handle words with multiple senses.

The Transformer architecture was a major breakthrough, speeding up training time by eliminating the need for RNNs while simultaneously improving the ability of language models to create contextualized vector representations. However, the Transformer in its most basic form is typically used for tasks such as translation, which, while immensely practical, are slightly less interesting than the implications this new architecture may have as a model of semantic memory. Thus, in the next section of this paper I will discuss the semantic implications of BERT: a new, cutting-edge language model which uses the Transformer architecture as a base.

Bidirectional Encoder Representations from Transformers (BERT)

BERT (Devlin, Chang, Lee, & Toutanova, 2019) is currently the state-of-the-art model used in the NLP field and was successful enough to be integrated into Google's search algorithm—specifically to handle queries requiring more in depth natural language and conversational knowledge. Not only is BERT able to obtain groundbreaking results on common NLP benchmarks, but it is also able to perform word sense disambiguation better than other models through the use of contextualized representations. BERT makes use of the Transformer architecture (Vaswani et al., 2017) but is not the same as the vanilla architecture described above. BERT is a language model while the Transformer model was originally used for machine learning tasks. Since the goal of BERT is not to translate sentences, only the Encoder mechanism is used. Another difference between BERT and the vanilla Transformer is that BERT is a bidirectional model—allowing it to learn word contexts based on all surrounding words, both left and right. The other DSMs discussed in this paper and other models which use the Transformer architecture, like OpenAI GPT (Radford, Narasimhan, Salimans, & Sutskever, 2018), are unidirectional. OpenAI GPT specifically is a left-to-right architecture, meaning every new token passed to the model can only attend to previously processed tokens within the attention layers. The unidirectional restriction is not optimal when processing input at the sentence level not only for practical purposes but for considerations of cognitive plausibility as well. When reading, humans can remember and understand the relationships between words in the beginning and end of a sentence, while unidirectional models struggle to do so. Thus, the true bidirectional processing ability of BERT is a huge advantage when it comes to concerns of practicality and cognitive plausibility.

BERT is trained in two steps: a pre-training phase and a fine-tuning phase. During the pre-training phase, BERT is trained over unlabeled data via two strategies: the Masked Language Model (MLM) and the Next Sentence Prediction (NSP) objectives. Then, after the model is pre-

trained, it is fine-tuned for a specific NLP task using labeled data from the specific task. Each new NLP task has different fine-tuned versions of the model. As a cognitive scientist, my interest in BERT is primarily in the methods used to pre-train the model and their implications for human learning. Thus, the methods used to fine-tune the model will not be discussed further as they are less informative of general semantic learning. Additionally, there is minimal difference between the pre-trained BERT model and the fine-tuned model (Devlin et al., 2019) making a discussion of both somewhat redundant.

To begin the pre-training phase, BERT is trained using the MLM objective. Under the MLM training strategy, 15% of input tokens are replaced with a MASK token. The goal is to predict the original vocabulary word based on the surrounding context of the MASK token. The MLM strategy is what makes BERT a true bidirectional model. Usually, a bidirectional model would be the result of concatenating representations from both a left-to-right model and a right-to-left model. This is because models are typically limited to either left-to-right or right-to-left architectures, since bidirectional training could allow words to "see themselves", allowing a model to trivially predict the target word in context. However, the MLM training strategy, by masking certain input tokens, allows BERT to be trained bidirectionally without being able to trivially predict the masked word based on context. After completing the MLM training phase, the model is trained using the NSP strategy. The goal of NSP is to train a model that is able to understand the relationship between two sentences. The training data is easily created from any monolingual corpus and is presented to the model as pairs of sentences: sentence A and sentence B. The model is then tasked with predicting whether sentence B follows sentence A; 50% of the time sentence B does follow sentence A and 50% of the time sentence B is a random sentence from the corpus. While this is a very simple training strategy, Devlin et al. (2019) found that including this step in

the pre-training phase allowed BERT to perform significantly better on inference tasks such as question answering.

Both pre-training strategies employed by BERT are fundamentally an implementation of predictive learning. In fact, the goal of the MLM training strategy is similar to the CBOW training strategy employed by word2vec: to predict a target word given context. Though both training methods are predictive in nature, there are of course substantial differences between them. First, word2vec is considered a unidirectional model. Thus, word2vec is at a disadvantage compared to BERT when it comes to keeping track of contexts. As discussed before, the word embeddings created by word2vec are static, which prevents the model from handling words with multiple senses correctly. BERT on the other hand, is able to produce contextualized representations through the use of Multi-Head Attention modules which greatly improved performance on many NLP tasks including word sense disambiguation.

While the Multi-Head Attention modules are responsible for the contextualized representations produced by BERT and seem to serve a similar function to human attention when learning, the actual mechanism itself is not well understood. A commonly held view in the NLP field is that the attention mechanism is an important way to provide interpretability to the predictions made by a model. However, there is some work within the NLP field which claims that the attention weights are not interpretable and changing them has no significant effect on model prediction (Jain & Wallace, 2019) and contradictory work that claims that attention captures several interpretable linguistic concepts (Vig & Belinkov, 2019). Vashishth, Upadhyay, Tomar, and Faruqui (2019) show that attention weights are interpretable and correlate with feature importance measures when the attention weights are essential for the model's prediction—which is not always the case. In fact, Kovaleva, Romanov, Rogers, and Rumshisky (2019) found that if attention heads were disabled one at a time, the model performance did not decrease and actually

increased in some cases. This suggests that the model may contain duplicate information in multiple attention heads and may be overparameterized. As of now, the nebulousness of the attention mechanism makes it fairly difficult to understand the exact implications the mechanism has for human learning—indicating a need for such work in the future. Regardless, the contextualized representations used by BERT have resulted in remarkable performance on NLP benchmark tasks as well as a leap in word sense disambiguation performance compared to its static representation competitors (Ethayarajh, 2019).

General Discussion

When debating the advantages and disadvantages of DSMs, cognitive scientists often focus on the learning mechanisms employed by each model. Indeed, the learning mechanism is important to the theoretical implications of each model and typically, there is some theoretical support for the learning mechanisms in question. Error-driven learning is supported by classic theories of reinforcement learning (e.g., Rescorla & Wagner, 1972), passive co-occurrence learning is supported by principles of associative learning (e.g., Murdock, 1982), and semantic abstraction at retrieval is supported by exemplar theories from the categorization literature (e.g., Nosofsky, 1986). In this paper, however, I have taken a different approach to comparing and contrasting DSMs. Instead of focusing on and debating between learning mechanisms I have focused on the implications that sequential learning and contextualized representations vs. static representations have for human semantic memory. The ability to learn sequentially seems like an absolutely necessary feature to constrain DSMs to human mechanisms of learning. The psychological implications of models which can't learn sequentially are too outlandish, particularly for a model of semantic memory. These models suggest a form of learning which requires all episodic memories to be present before any abstraction to semantic memory can occur. This is clearly not

the case in humans as we have access to semantic memory long before the end of our lives. Fortunately, most modern models have incorporated the ability to learn sequentially making further discussion of its necessity moot.

While the ability to learn sequentially is practically ubiquitous in modern DSMs, each model faces its own unique challenges in doing so. One such challenge faced by predictive neural networks is catastrophic forgetting—the tendency to lose one set of associations when learning new ones. CF is unique to predictive neural networks and is caused by the specific learning mechanism employed by the model. CF is exposed in predictive DSMs, like word2vec, when attempting to learn representations for words with multiple senses considering. While CF is not a problem in other DSM architectures, like the RVA which do not use an error-signal to learn, these models are still unable to handle words with multiple senses correctly due to the static nature of the vector representations they produce.

Many models that I've discussed in this paper are very successful at modeling and fitting psychological data and use static representations, so why should contextualized vectors be a necessary constraint for DSMs? The fact of the matter is, humans are able to effectively understand and perform word sense disambiguation while DSMs have struggled with this task since conception. The ability of DSMs to perform word sense disambiguation hit a brick wall (de Lacalle & Agirre, 2015) before contextualized vectors were widely introduced with the Transformer model and BERT. Contextualized vectors are not only superior to static vectors when it comes to NLP benchmarks, but they allow DSMs to more closely model human language ability. While contextualized vectors are fairly new, I introduced and discussed two models which make use of contextualized vectors: the exemplar-based DSM and the Transformer/BERT models. These models are very different in their origins. The exemplar-based DSM originated from the cognitive

science field and has very clear implications for human semantic memory, yet it suffers from memory overflow and is unable to scale up to the huge datasets commonly used in for training modern DSMs. On the other hand, the Transformer/BERT model originated from the machine learning and NLP fields and while it is easily able to handle the large natural language datasets, there have been relatively few studies investigating the cognitive implications of this model architecture. The deficits of each model suggest an interdisciplinary approach between cognitive science and NLP is necessary to really make use of and understand models that use contextualized vector representations.

Conclusion

Throughout this paper, I have reviewed and discussed several traditional and modern DSMs. The discussion of the traditional DSMs served to highlight the importance of sequential learning while the discussion of the modern DSMs served to highlight the importance of contextualized representations when it comes to word sense disambiguation. Additionally, each model has unique implications for human semantic memory and learning which have been discussed throughout. By considering the implications and abilities of each model, I aim to support the claim that sequential learning and contextualized representations are two necessary constraints on human semantic learning that any future research into DSMs should take into consideration.

References

Asr, F. T., & Jones, M. N. (2017). *An Artificial Language Evaluation of Distributional Semantic Models*. Paper presented at the 21st Conference on Computational Natural Language Learning (CoNLL).

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 1*, 238-247.

Benigo, Y., Lee, D. H., Bornschein, J., Mesnard, T., & Lin, Z. (2015). Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*.

Benna, M. K., & Fusi, S. (2015). Computational principles of biological memory. *arXiv preprint arXiv:1507.07580*.

de Boer, J. N., Voppel, A. E., Begemann, M. J. H., Schnack, H. G., Wijnen, F., & Sommer, I. E. C. (2018). Clinical use of semantic space models in psychiatry and neurology: A systematic review and meta-analysis. *Neuroscience & Biobehavioral Reviews, 93*, 85-92.

de Lacalle, O. L., & Agirre, E. (2015). A methodology for word sense disambiguation at 90% based on large-scale crowdsourcing. *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, 61-70.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT*(1).

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

Elman, J. L. (1995). Language as a dynamical system. *Mind as motion: Explorations in the dynamics of cognition*, 195-223.

Ethayarajh, K. (2019). How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. *arXiv preprint arXiv:1909.00512*.

Firth, J. R. (1957). *A synopsis of linguistic theory.* Oxford.

Foltz, P. W. (1996). Latent Semantic Analysis for text-based research. *Behavior Research Methods, Instruments, & Computers, 28*, 197-202.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., & Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural netowrks. *arXiv preprint arXiv:1312.6211*.

Grossberg, S. (1987). Competitive learning: from interactive activation to adaptive rsonance. *Cognitive Science, 11*, 23-63.

Gunther, F., Rinaldi, L., & Marelli, M. (2019). Vector-space models of semantic representation from a cognitive perspective: A discussion of common misconceptions. *Perspectives on Psychological Science, 14*(6), 1006-1033.

Harris, Z. (1954). Distributional structure. *Word, 10*(2-3), 146-162.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Jain, S., & Wallace, B. C. (2019). Attention is not explanation. *arXiv preprint arXiv:1902. 10186*.

Jamieson, R. K., Johns, B. T., Avery, J. E., & Jones, M. N. (2018). An instance theory of semantic memory. *Computational Brain & Behavior, 1*(2), 119-136.

Johns, B. T., Mewhort, D. J. K., & Jones, M. N. (2019). The role of negative information in distributional semantic learning. *Cognitive Science, 43*(5), e12730.

Jones, M. N. (2017). *Big data in cognitive science*: United Kingodm: Psychology Press.

Jones, M. N. (2018). When does abstraction occur in semantic memory: insights from distributional models. *Language, Cognition, and Neuroscience*, 1-9.

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

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

Kanerva, P. (2009). Hyperdimensional computing: An introduction to computing in distributed representations with high-dimensional random vectors. *Cognitive Computation, 1*, 139-159.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., . . . Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedngs of the national academy of sciences, 114*(13), 3521-3526.

Kovaleva, O., Romanov, A., Rogers, A., & Rumshisky, A. (2019). Revealing the dark secrets of BERT. *arXiv preprint arXiv:1908.08593*.

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), 211-240.

Lopez-Paz, D., & Ranzato, M. A. (2017). *Gradient episodic memory for continual learning.* Paper presented at the Advances in neural information processing systems.

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from 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.

Mannering, W. M., & Jones, M. N. (2020). Catastrophic Interference in Predictive Neural Network Models of Distributional Semantics. *Computational Brain & Behavior*. doi:https://doi.org/10.1007/s42113-020-00089-5

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review, 102*(3), 419-457.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation, 24*, 109-165.

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., . . . Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529-533.

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

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

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wertmer, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks, 113*, 54-71.

Perfetti, C. A. (1998). The limits of co-occurrence: Tools and theories in language research. *Discourse Processes, 25*, 363-377.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding with unsupervised learning. *Technical report, OpenAI*.

Ramscar, M., & Yarlett, D. (2003). Semantic grounding in models of analogy: An environmental approach. *Cognitive Science, 27*(1), 41-71.

Ratcliff, R. (1990). Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review, 97*, 285-308.

Reed, S. K. (1972). Pattern recognition and categorization. *Cognitive Psychology, 3*, 382-407.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory, 2*, 64-99.

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.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533-536.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J. L. (1991). Graded state machines: The representation of temporal contingencies in simple recurrent networks. *Machine Learning, 7*(2-3), 161-193.

Stanton, R. D., Nosofsky, R. M., & Zaki, S. R. (2002). Comparisons between exemplar similarity and mixed prototype models using linearly seperable category structure. *Memory & Cognition, 30*, 934-944.

Vashishth, S., Upadhyay, S., Tomar, G. S., & Faruqui, M. (2019). Attention interpretability across nlp tasks. *arXiv preprint arXiv:1909.11218*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 5998-6008.

Vig, J., & Belinkov, Y. (2019). Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*.