METAPHOR AND SYMBOL
2018, VOL. 33, NO. 4, 280–294
https://doi.org/10.1080/10926488.2018.1549840

Vector Space Applications in Metaphor Comprehension

J. Nick Reid and Albert N. Katz

Western University

ABSTRACT

Although vector space models of word meaning have been successful in modeling many aspects of human semantic knowledge, little research has explored figurative language, such as metaphor, using word vector representations. This article reviews the small body of research that has applied such representations to computational models of metaphor. After providing a short review of vector space models, a detailed overview of metaphor models that make use of vector space, and the relevant empirical findings are discussed. These models are divided into two categories based on their differing motivations: "psychological" models are motivated by modeling the cognitive processes involved in metaphor comprehension whereas "paraphrase" models seek to find the most efficient and accurate way for a computer to paraphrase metaphorical language. These models have been successful in computing adequate metaphor interpretations and shed light on the cognitive processes involved in comprehending metaphor.

There are two main streams in cognitive approaches to understanding how metaphors are recognized, comprehended, and interpreted. The cognitive linguistic approach, starting with the seminal work of Lakoff and Johnson (1980), posits that the motivation for understanding metaphor resides not in the linguistic metaphoric expression per se, but in underlying root or conceptual metaphors, and the relations among these conceptual metaphors. Thus the underlying mental work in understanding even a nominal metaphor (e.g., *"Juliet is the sun"*) resides in the activation of a larger cross-concept mapping (e.g., *"PEOPLE ARE ASTRONOMICAL OBJECTS"*). In contrast, there is an older tradition traced back to Aristotle's seminal works, *Poetics* (1996) and *Rhetoric* (1952), in which metaphor is understood through the stretching of the meaning of the words in the linguistic expression itself, such as finding a way that the meaning of the word "sun" can be applied to the word "Juliet," to produce a meaning not totally explicable by the meaning of the two words when not juxtaposed. The aim of this article is to review recent advances in computational linguistics based on "vector space models" and their applications to the understanding of metaphor recognition, comprehension and interpretation, most of which is based on the second, Aristotelian tradition.

Computational linguistics is an interdisciplinary area of research that seeks to understand elements of language by analyzing statistical regularities within text corpora. In particular, it draws on knowledge from linguistics, cognitive science, and computer science, and models language using tools from mathematics such as geometry, linear algebra, and probability. A subset of computational linguistics models is known as "vector space models," which represent words as vectors in a high-dimensional space. Similarity between words is then calculated based on either the distance or the cosine of the angle between the words' vectors. Typically the vectors are created based on the co-occurrences of words or the occurrences of words within a context, such as a set of sentences, paragraphs, or documents.

Vector space models have many applications. They are particularly useful for automatic indexing and information retrieval, relevant to issues in computer science (see Deerwester, Dumais, Furnas, Landauer, &

CONTACT J. Nick Reid jreid256@uwo.ca Department of Psychology, Western University, Social Science Centre, 1151 Richmond St., London, Ontario N6A 5C2, CANADA

Harshman, 1990; Hofmann, 1999). They can also model how humans acquire language, which is more relevant to the aim of the current article, the processing of metaphor. For instance, Latent Semantic Analysis (LSA), a popular vector space model, performs similarly to second-language English learners on a synonym selection test, with both LSA and the sample correctly answering about 65% correct (Landauer & Dumais, 1997). Additionally, vector similarity values correlate with reactions times from human lexical priming experiments (Landauer & Dumais, 1997; Lund & Burgess, 1996). Google's Word2vec model can actually compute what dimensions are similar between words, and can use this information to solve both syntactic and semantic analogies, such as "big" to "bigger" equals "cold" to "colder" and "king" to "queen" equals "man" to "woman" (Mikolov, Chen, Corrado, & Dean, 2013a; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013b). To understand the application of these models to metaphor, it is necessary to first provide a short primer on the basic assumptions and types of vector space models.

Basic assumptions and types of vector space models

All vector space models of word meaning are based on the "distributional hypothesis," the assumption that the meaning of words can be understood by examining the contexts they appear in and the other words they appear with (Harris, 1954; Rubenstein & Goodenough, 1965; Sahlgren, 2008). This idea was first articulated by Harris (1954) who gave examples of how both morphologies and meanings of words could be ascertained based on their contexts; for example, if two words appear in very similar contexts but the two words themselves never appear in the same sentence, they are likely to be synonyms (p. 157). Although Harris' paper was only theoretical and there was not yet the computing technology to perform calculations on large-scale corpora, it touched on many of the important issues that would eventually be relevant to the vector space models of today. Specifically, Harris argued that the semantic difference between the meaning of two words would be related to the amount of difference in their contexts (p. 157).

Recent approaches have implemented the seminal ideas of Harris (1954) operationalizing and representing word contexts, and by examining their assumptions using large corpora (Clark, 2015). An example of this notion follows. Consider a matrix consisting of a set of words (in the rows), and the various contexts in which they might occur (in the columns) for the following three sentences:

(1) Cognitive psychology is the study of fundamental mental processes, such as memory and language.
(2) In contrast to cognitive psychology, behavioral psychology is concerned with finding laws that govern behavior.
(3) Today, neuroimaging techniques are used in psychology to observe brain activation associated with certain cognitive tasks.

In this case, the columns of the matrix representing the contexts correspond to the three sentences (S1, S2, S3) and a selection of words make up the rows (e.g., cognitive, psychology, processes). The numbers in the matrix are the occurrence counts of each word in each context. This generates the following matrix:

	S1	S2	S3
Cognitive	1	1	1
Psychology	1	2	1
Processes	1	0	0
Govern	0	1	0
Behavior	0	1	0
Activation	0	0	1

Admittedly even in this extremely simplified example, the matrix shown allows for some simple calculations such as the dot product between two vectors (Clark, 2015). For example, the dot product of the word vectors for cognitive [1, 1, 1] and psychology [1, 2, 1] equals 4 $((1 \times 1) + (1 \times 2) + (1 \times 1) = 4)$ whereas for cognitive [1, 1, 1] and govern [0, 1, 0], the dot product equals 1. Comparing the dot products of the two pairs would give a very simplistic comparison of similarity between the terms; that is, the greater dot product for the vectors associated with the words "cognitive" and "psychology" would indicate that the meaning of those two words are more similar to one another than is the meaning shared by "cognitive" and "govern," whose vectors have a smaller dot product. Naturally, an overly simplistic application of this procedure is problematic for several reasons: (a) some words occur much more frequently than others (e.g., function words such as "the" and "a") and their co-occurrence with other words may not be informative in similarity comparisons between concepts, (b) each context will use only a fraction of the words in the language, leading to many zero occurrence counts, and therefore, underestimation of similarity, and (c) words that are highly similar in meaning, such as synonyms, may occur in similar contexts, but the synonym pair itself is unlikely to co-occur because the author is unlikely to use both terms in the same sentence, again leading to an underestimation of word meaning similarity.

To address these issues, most word vector models manipulate the matrix in three ways. First, weights are typically applied to the vectors so that the more informative words are weighted more heavily (Durda & Buchanan, 2008; Harman, 1986; Landauer & Dumais, 1997). Second, the vectors are either normalized to a constant length (Lund & Burgess, 1996) or similarity between word vectors is computed as the cosine of the angle between the vectors (Clark, 2015; Landauer & Dumais, 1997). Because some word vectors have greater length, measuring the distance between vectors may underestimate similarity—the distance may be large simply because one word occurs more frequently than the other. Thus, normalizing the vectors or computing the cosine between them more accurately captures the true similarity between words. Third, typically the matrix is reduced to a smaller set of dimensions (Hofmann, 1999; Landauer & Dumais, 1997). The logic here is that little information is provided by any two words co-occurring on their own; what is required is the extraction of deeper semantic information about the relations between words. The process of reducing dimensions is similar to finding a set of words that all exemplify a common topic or theme. This step solves the issue of zero occurrence counts, because even if two related words do not co-occur, they likely both occur in the same general context, which is more likely captured in the reduced dimension matrix. The actual methods that are used to reduce dimensions differ with specific vector space models.

The LSA model

LSA records the occurrences of words within contexts (typically documents, although sometimes paragraphs or a large span of words is used) and therefore uses a large context window (Deerwester et al., 1990; Landauer & Dumais, 1997; Landauer, Foltz, & Laham, 1998). LSA uses a "bag-of-words" approach in that only the raw occurrences are counted without any notion of syntax, logic, order, or proximity. One of the defining features of LSA is its use of Singular Value Decomposition, which is a mathematical technique that reduces a matrix to a much smaller set of principle components and, as such, closely resembles factor analysis (Landauer & Dumais, 1997). The effect is to extract a number of meaning dimensions (or "latent" concepts) that are shared by groups of words and documents. Once these dimensions have been extracted, they can be treated as contexts, and then words can be compared to see if they frequently co-occur in similar contexts. It should thus be noted that the contexts in the reduced matrix are mathematical abstractions and cannot be easily interpreted (Hofmann, 1999; Kintsch, 2000). Typically 300 meaning dimensions are retained for simulations with LSA as this yields optimum performance on synonym judgment tasks (Landauer & Dumais, 1997), a clear measure of meaning similarity.

Dependency-based vector space models

Dependency-based vector space models consider syntactical relations in constructing the semantic space (Lin, 1998; Padó & Lapata, 2007). Padó and Lapata (2007) offered a general framework for constructing a dependency-based semantic space by adding some additional parameters to the matrix construction. The first major difference from LSA is that the matrix is based on the co-occurrences of words (i.e., a word-by-word matrix; see Lund & Burgess, 1996) rather than the occurrences of words in contexts. Also, unlike LSA's bag-of-words approach, which disregards syntax, co-occurrence counts in dependency-based models are based on syntactic "paths." For instance, consider the example sentence: "a lorry might carry sweet apples." Syntactically "apples" is the direct object of "carry," and therefore the path between "carry" and "apples" (carry –> apples) is one. In contrast, the path between "carry" and "sweet," although directly adjacent, is actually longer (two) because these two words are not in a direct syntactic relationship (carry –> apples –> sweet; the object being carried is sweet, so "carry" and "sweet" are only connected based on the object being tied to both). A path of three obtains when the co-occurrence between two words is based on three paths. Thus, words are more similar in meaning the fewer the number of paths and are less similar as the number of paths increases. For simulation purposes, optimum results occur when co-occurrence counts are restricted to words connected by three or fewer paths. The model also permits for the weighting of paths; for instance, certain types of syntactic relations can be weighted heavier, such as subjects and objects of verbs, when they might be more informative than other types of relations. Also, co-occurrences can be weighted inversely proportional to the number of paths between the words, such that co-occurrences of words that share a direct syntactic relation are weighted heavier than those that share a more indirect relation. After the weights are applied, the end result is a word-by-word co-occurrence matrix, although other dependency-based models sometimes count the co-occurrences of words along with their relations. For instance, Lin's (1998) model consists of occurrence counts for "dependency triples" (e.g., [apples, object-of, carry]). Although this is optional, Padó and Lapata do not include the syntactic relations in their simulations as the simpler word-by-word matrix permits comparison to more traditional word-based models.

One advantage of this approach is that co-occurrence counts are based on direct relationships between words in sentences, and therefore, the semantic relatedness of the words cannot be disputed. Of course, this also adds computational complexity and requires some form of syntactic annotations, but Padó and Lapata (2007) argued the additional complexity is worth the improvement in simulations. Compared to LSA, dependency-based models capture a tighter, synonym-like similarity whereas LSA's larger context window captures a broader, topical similarity (Mohler, Rink, Bracewell, & Tomlinson, 2014; Van de Cruys, Poibeau, & Korhonen, 2011).

Word2vec model

Word2vec is based on predicting what words are likely to occur in a context-window given the other known words in that window. The vectors generated through this predictive process capture a type of similarity somewhere between those obtained by LSA and dependency-based models (Mohler et al., 2014). Word2vec was developed by Mikolov et al. (2013a, 2013b) and consists of two separate models: Continuous-bag-of-words and Skip-gram. Both use neural networks, computer models that, arguably, simulate how the human brain learns and processes information (Agatonovic-Kustrin & Beresford, 2000; Rumelhart, Hinton, & Williams, 1986).[1]

[1]Neural networks operate by detecting patterns in data and learn from exemplars (or experience) rather than a set of programmed rules. Neural networks usually have at least three layers of processing units (or neurons): an input layer, a hidden layer, and an output layer. The input layer receives the data and the output layer provides some response to this data, which can vary depending on the processing task. The information is passed from input to output through a hidden layer, which is a set of neurons that further process the information. This layer can be thought of as a set of learned features that the model deems important for predictions (Touretzky & Pomerleau, 1989). One of the key components of neural networks are the weights between neurons, which determine which connections between neurons should be weighted more heavily in predictions. The weights are adjusted as the model is trained on exemplars through a process called "backpropagation," in which the output of the model is compared with the desired response (Rumelhart et al., 1986). After each exemplar, the error is fed back through the model and the weights are adjusted to minimize the error. For example, when the connection between two particular neurons contributes to the error, the weight between them is decreased, thus improving the model. This is how the model learns from experience—with each training exemplar the model recalibrates to provide a better prediction. Typically once the model reaches some criteria of accuracy, the weights are recorded and are used to make predictions on a new set of exemplars.

With Continuous-bag-of-words, the task of the neural network is to predict a target word based on the words that surround it. For example, given a training exemplar such as "the horse raced past the barn fell," the input would be the words "the," "horse," "raced," "the," "barn," and "fell," and the desired output would be the word "past." The input layer has a neuron for each word in the vocabulary and each of these neurons has a weighted connection to each of the hidden layer neurons, which represent meaningful dimensions or features, somewhat similar to the latent dimensions in LSA (Levy & Goldberg, 2014). These weighted connections form a "weight matrix" in which the rows are words, the columns are hidden layer neurons (or meaning dimensions), and the values are the weighted connections between the words and hidden layer neurons. The output layer also has a neuron for each word in the vocabulary, and each neuron in the hidden layer has a weighted connection to each output layer neuron, which forms a second weight matrix. The output layer neurons that receive the most activation correspond to the words the model predicts are most likely to occur. As with most neural networks, the model is trained on many exemplars (in this case, windows of words taken from a large text corpus) and weights are adjusted after each prediction to bring the output closer to the desired response. Similar to LSA, the model uses a "bag-of-words" approach (i.e., the order of the inputted words is irrelevant).

For the Skip-gram model, the task of the neural network is almost opposite—the model predicts the surrounding words of a single inputted word (Mikolov et al., 2013a, 2013b). So, for the sentence "the horse raced past the barn fell," the model may receive the word "past" as input and the desired output would be the rest of the words in the sentence, or more accurately, context window. The neural network operates in a similar way, except that the input layer just receives a single word vector and the output layer makes predictions for multiple words (i.e., the word two spaces before the target, one space before, one space after, etc.). The rest of the neural network architecture is the same; there are input and output neurons corresponding to all words in the vocabulary and hidden layer neurons corresponding to meaning dimensions.

The main purpose of Continuous-bag-of-words and Skip-gram is to generate word vectors, not to make predictions of text. Therefore, the prediction tasks are simply mechanisms for learning weights between words and dimensions, and these learned weights are what make up the final vector representations. Thus, the word vectors from Word2vec do not come directly from a co-occurrence matrix, but they are based on co-occurrence because the weights that make up the matrix are calculated by maximizing predictions of co-occurrence. This is a subtle distinction, but it means the matrix is not formed based on occurrence counts like the other models reviewed. An advantage of this approach is that a very large matrix does not have to be constructed first and then reduced. Instead, the dimensions are learned iteratively, which is less computationally costly.

Vectors space models and metaphor comprehension

In the Aristotelian tradition of metaphor processing, the linguistic terms juxtaposed in a metaphor are labeled the "topic" (or tenor) and "vehicle." The topic is the term that is being framed or talked about whereas the vehicle is the term used to frame the topic; for instance, in the metaphor *"my lawyer is a shark,"* "lawyer" is the topic and "shark" is the vehicle. The difficulty in modeling metaphor is that the topic and vehicle are mostly dissimilar with only a handful of features relevant to apply sharks to lawyers in a meaningful fashion (Bowdle & Gentner, 2005; Glucksberg & Keysar, 1990). Computationally, this means that simply calculating the centroid (taking the average) between the word vectors for the topic and vehicle is inadequate as most of the associates of the vehicle are not relevant to the metaphor (Kintsch, 2000). Furthermore, metaphor is asymmetrical as the relevant features from the vehicle are projected onto the topic, therefore, reversing a metaphor (e.g., *"a shark is a lawyer"*) leads to a different or nonsensical meaning (but see Katz & Al-Azary, 2017, for conditions that promote bi-directionality). Computational models of metaphor must capture these critical aspects to be adequate. Early attempts used proto-vector non-computational approaches to metaphor, where words in multidimensional space were based on the factor analyses of attributes known a priori to be relevant to metaphor appreciation and comprehension (e.g., Tourangeau & Sternberg, 1981; Trick & Katz, 1986).

In the pre-vector computational models of metaphor, hand-coded inputs were required from the researcher. Fass (1991) developed Met*, a model that automatically classified statements into four categories: literal, metaphor, metonymy, and anomalous. The model worked by evaluating "preferences," for example, the verb "drink" has a preference that the agent for this verb be an animal because animals drink. In a statement such as *"the car drank gasoline,"* the agent "car" violates the preference of the verb "drink." For metaphor, preference is violated, but a relevant analogy is present. Met* finds relevant analogies by searching the sense-networks of both the term that violates the preference (e.g., "car") and the preferred term itself (e.g., "animal") for any "sister" relations, which is when two terms are members of a higher order category. For the above example, Met* finds that animals "drink" and cars "use," drink and use both being members of an "expend" category. Because there is a relevant analogy (cars expend gasoline like animals expend liquids they drink), *"the car drank gasoline"* is classified as a metaphor.

Falkenhainer, Forbus, and Gentner (1989) developed the structure-mapping engine (SME), which implemented Gentner's (1983) structure-mapping theory of analogy, the core of which is the "systematicity principal" (i.e., preference to find similarity in relations between two domains rather than similarities of attributes). The preferred analogy (or metaphor) is a consistent system of relations between two domains. For example, for the analogy *"an atom is a solar system,"* the nucleus is mapped onto the sun and the electrons are mapped onto planets, which maintains the relation that electrons "rotate" around the nucleus as planets "rotate" around the sun. The model will prefer this higher order mapping over a shallower mapping that aligns individual features (e.g., the sun and a nucleus are both "round").

Although Met* and SME obtained high accuracy in classification and interpretation, respectively, the major limitation was that they required large amounts of manually coded information. As a result, the models could only be tested on a limited amount of metaphors and could not generalize to new metaphors. Additionally, manually coded information is susceptible to experimenter bias, wherein researchers might unintentionally input features that skew the models toward metaphor interpretation.

Although the models mentioned above were limited, they laid the foundation for modern computational models that use vector space word representations. The main advantage of these newer models is that they do not require as much hand-coding by the researcher and can easily be tested on new sets of metaphor stimuli. The metaphor comprehension models that have applied vector space representations fall into two general categories, which we will refer to as "psychological" and "paraphrase" models. The psychological models are focused on modeling proposed theoretical cognitive processes of metaphor comprehension found in the archival literature, such as categorization (Glucksberg & Keysar, 1990) and comparison (Bowdle & Gentner, 2005; Gentner, 1983). These models involve algorithms that compute a new vector representing the metaphor's meaning, and the algorithm operates on the semantic space alone, with no other knowledge source involved. The metaphor vector is then compared to human generated interpretations to assess how closely the responses output by the algorithm resemble human responses. In contrast, the goal of paraphrase models is to generate the most accurate and plausible literal paraphrase for a given metaphor. These models come from the field of Natural Language Processing (NLP) and therefore are not as concerned with the psychological process, but rather finding the most accurate way for a computer to interpret metaphorical language. Vector-based paraphrase models usually use vector representations along with other data-mining techniques to maximize paraphrase accuracy. Rather than comparing the paraphrases to human-generated interpretations, usually the paraphrases are assessed directly by a group of human raters.

"Psychological" metaphor models

Kintsch's predication algorithm

Kintsch (2000) described the first metaphor comprehension model that made use of word vectors. One of the major issues with metaphor comprehension is finding the relevant features of the vehicle that apply to the topic. Kintsch's algorithm does this by searching the semantic neighborhood of the vehicle and selecting the word vectors that are at least somewhat related to the topic. In particular, it searches the 500–1,500 nearest neighbors of the vehicle, as calculated by LSA, and selects a set number of words (five in Kintsch's simulations) that have the highest cosines with the topic. The size of the neighborhood is critical because metaphor involves a comparison of two unlike things, and if too many or too few words are considered, one faces the likelihood irrelevant features will be selected.

Once the words from the vehicle's neighborhood are selected, the centroid between these words, the topic, and the vehicle is computed. This computation has been compared to Glucksberg and colleagues' categorization model of metaphor comprehension as the centroid of the selected terms along with the topic and vehicle can be thought of loosely as an *ad hoc* category of which the topic is a member (Glucksberg, 2003; Kintsch, 2000). However, the model also resembles Ortony's (1979) salience imbalance model, as the features selected from the vehicle's neighborhood are of higher salience to the vehicle and lower salience to the topic. This calculated centroid vector is interpreted by comparing it to other landmarks in the semantic space.

Kintsch (2000) provides an example of how the computed vector for the metaphor *"my lawyer is a shark"* compares to the word vector for "lawyer" (see Kintsch, 2000, p. 260). Lawyer terms such as "justice" and "crime" have about equally high cosines (~.6) with both vectors; however, the critical term "viciousness" increases from about .12 for the lawyer vector to .25 for the metaphor vector.[2] In this way, the algorithm has highlighted this feature as relevant to the metaphor. However, the metaphor vector is also closer to irrelevant fish-type terms such as "shark" and "fish"; these cosines increase from 0 to about .12. Although the algorithm correctly emphasized "viciousness," one obvious issue is that both "justice" and "crime" have higher cosine similarity to the metaphor vector than does "viciousness." Presumably these terms are also irrelevant to the metaphor, or at least less related than viciousness. A shark obviously does not elicit thoughts of crime or justice, but the cosines remain high because they are related to lawyer. Thus, if the calculated metaphor vector truly captured the metaphor's meaning, it should have lower cosines with these words than it does with viciousness. Furthermore, the landmarks mentioned above (i.e., justice, crime, viciousness, etc.) were hand-selected, so it is unclear how many words have cosines higher with the metaphor vector than viciousness (for instance, other irrelevant lawyer-type words like "jury" and "courtroom" may also have higher cosines than viciousness). Looking at only the magnitude of change is inappropriate as well because "shark" and "fish" were emphasized about equally to "viciousness." For these reasons, this model is weaker than newer models that provide more definitive interpretations. Nonetheless, this model generates vectors that more closely resemble participant-generated interpretations of metaphors than topic or vehicle vectors alone (Kintsch & Bowles, 2002). The model was also influential; both Utsumi's (2011) and Terai and Nakagawa's (2012) models adopted Kintsch's predication algorithm.

Utsumi's categorization and comparison algorithms

Utsumi (2011) used computational algorithms to model two separate hypothesized metaphor comprehension processes, categorization and comparison, and to explore what conditions promote one process over the other. As mentioned above, Kintsch's (2000) predication algorithm resembles

[2]Cosines are similar to correlation coefficients in that they vary in strength from 0 to 1 with larger values representing a stronger relationship. However, unlike correlation, negative cosine values are rare.

categorization, so Utsumi used this algorithm to simulate the categorization process. For the comparison process, Utsumi created an algorithm that searches the semantic neighborhoods of the topic and vehicle simultaneously for words common to both, starting from nearest neighbors proceeding to most distant. Once a set number of neighbors are found, a centroid between these words and the topic is calculated (unlike the categorization model, the vehicle is not included). There are subtle differences between the categorization and comparison algorithms: the categorization algorithm searches within the vehicle's semantic neighborhood for words that have high cosines with the topic whereas the comparison algorithm begins symmetrically as both the topic and vehicle's neighborhoods are searched for common words. Moreover, the categorization algorithm calculates the centroid between the selected words, the topic, and the vehicle because the topic and vehicle are both members of the ad hoc category. In contrast, the comparison algorithm involves projection only of elements connected to the shared relation of topic and vehicle and thus the comparison centroid is calculated from only the selected words and the topic.

Utsumi (2011) demonstrated that the algorithms generate very different vectors by conducting simulations on 40 different metaphors and then comparing the generated metaphor vectors to human interpretation. He found that the categorization algorithm best modeled human interpretations for 11 metaphors whereas the other 29 were best modeled by the comparison algorithm. Both conventionality and interpretive diversity were critical variables for predicting which model would perform best for a given metaphor. Interpretive diversity is a measure of semantic richness wherein diversity is high for metaphors in which participants generate many interpretations and each interpretation is about equally as frequent. In contrast, if few interpretations are generated and a single interpretation is dominant, interpretive diversity is low. The simulations showed that the comparison algorithm performed better with less conventional metaphors that were high in diversity. Conversely, the categorization algorithm performed better with conventional metaphors low in diversity. These computational data suggest that humans may use a categorization process when interpreting highly conventional and semantically narrow metaphors, but use a comparison process for less conventional and semantically rich metaphors. To our knowledge this study was the first time a computational method was used to compare competing theoretical process models rather than merely simulating existing ones.

Terai and Nakagawa's two-process model

Terai and Nakagawa (2012) created a two-process model of metaphor comprehension that included categorization and dynamic interaction among features. The categorization process resembled Kintsch (2000), but the semantic space used in the model consisted of a noun-by-feature matrix rather than a word-by-context LSA matrix. To create this semantic space, Terai and Nakagawa extracted latent classes from adjective–noun and verb–noun pairs mined from a corpus of Japanese newspaper articles (see Kameya & Sato, 2005). This type of semantic space resembles a dependency-based model in which only certain types of syntactic relations are counted: adjective–noun (modification), noun (subject)–verb, verb–noun (modification), and verb–noun (object) relations. Once the latent classes are extracted, the probability of an adjective/verb (i.e., a "feature") given a noun can be estimated based on how strong of members the two terms are of the various latent classes.

In addition to the categorization process, Terai and Nakagawa (2012) included a novel dynamic-interaction process to identify emergent features (i.e., features elicited by the interaction between the topic and vehicle terms that are not elicited by either term alone). To model this process, Terai and Nakagawa used a recurrent neural network in which each hidden layer neuron has connections to all other hidden layer neurons, and when two of these neurons have a strong connection, they both become more activated. For the dynamic-interaction process, the input consisted of the metaphor vector calculated in the categorization process. As with the noun vectors in the semantic space, the computed metaphor vector consisted of a neighborhood of features, with the values representing the probabilities of the features given the metaphor. To select the features for dynamic interaction, a threshold was set so that only features with high probabilities were included. The hidden layer of the neural network consisted of a series of nodes with each one corresponding to one of the selected features. As mentioned above, each feature node has a connection weight with every other feature node. To estimate the connection weights, Terai and Nakagawa first constructed a "sibling neighborhood" of the 50 most related nouns to the topic and the 50 most related nouns to the vehicle. Next, the correlation between two features across the sibling neighborhoods was computed and used as the weight between that pair of nodes. This was done for all feature node pairs with the logic being that the features highly correlated with many of the other features (at least in terms of their relation with the sibling neighborhood nouns) would become more activated and would be weighted higher in the final outputted metaphor vector. Essentially, this neural network involved the inputted metaphor vector, a hidden layer of feature nodes that interacted with each other, and an outputted metaphor vector influenced by the interactions in the hidden layer, with the features highly correlated with other features being more emphasized. It was the interactions of the hidden layer nodes that were thought to capture the dynamic interaction of topic and vehicle in metaphor comprehension. Note that unlike most neural networks that adjust weights based on comparing the model's output to the desired output, the neural network used here instead estimated connection weights based on the correlation procedure mentioned above.

To assess if the dynamic interaction process actually captured emergent features, Terai and Nakagawa (2012) collected participant responses for several metaphors. One group was instructed to list features of just the topic and vehicle terms alone (not combined in a metaphor) whereas the other group was instructed to list features of the metaphor. Emergent features were defined as features listed only by the metaphor group and not by the topic/vehicle group. Terai and Nakagawa (2012) found that the model captured more emergent features when the dynamic-interaction process was included versus the categorization process alone. A group of participants also directly evaluated the model's output and rated the simulations that included dynamic interaction as more valid. The results of both the feature listing and evaluation tasks suggested that including the dynamic-interaction process significantly improved the model over the categorization process alone.

"Paraphrase" metaphor models

Shutova, Van de Cruys, and Korhonen's paraphrase model

In contrast to the previous models that compute new vectors to represent metaphors, Shutova, Van de Cruys, and Korhonen (2012) instead treated metaphor comprehension as a paraphrase task in which the meaning of the metaphor term is determined based on the context; that is, essentially, they treat metaphor as a form of word-sense disambiguation.

The model is based on Van de Cruys et al.'s (2011) method for disambiguating between senses of nouns. Both dependency-based and context-based co-occurrence models are used to create three separate matrices: a word-by-dependency relations matrix, a word-by-context words matrix, and a dependency relations-by-context words matrix. They then find a set of latent features that apply to all three matrices using a data reduction method in which the latent features can be interpreted probabilistically (non-negative matrix factorization).

In order to disambiguate between word senses, Van de Cruys et al. (2011) adapted the target noun vector based on the context. First, a vector for the noun aggregated across all contexts is used for the general meaning of the noun (similar to LSA, all senses of the word will be lumped together to form a single vector). This can be a vector taken from the word-by-dependency relations matrix or the word-by-context word matrix, or a combination of both (Van de Cruys et al. found that the combination method yields the best simulation results). Next, the context words occurring with the particular instance of the noun under consideration are used to compute a vector representing the context. The vectors for all of the context words are taken from the context word-by-dependency relations matrix and are averaged to form a single vector representing the context. Because non-negative matrix factorization can be interpreted probabilistically, the vectors can be transformed into the probability distributions of the various latent features given the noun or context words. The final step is to point-wise multiply the noun vector by the context vector, which will emphasize the important latent features of the specific context while deemphasizing the latent features not important to that context. For example, in a sentence such as "Jack is listening to a *record*," point-wise multiplying the "record" noun vector by the context vector will emphasize music-type features while deemphasizing bookkeeping-type features that are important for the general meaning of record but not for this particular instance. In order to select a valid paraphrase, the noun that has the closest vector to the point-wise product of the original noun ("record") and context vector is selected, which in this case may be a word such as "album." In this way, the model has computed the sense of the noun "record" in this particular context and substituted another word, "album," that maintains the sentence's meaning.

Shutova et al. (2012) applied Van de Cruys et al.'s (2011) technique to verb-based metaphors (e.g., "*reflect* concern"). A metaphor can be thought of as a special type of word sense in which the context emphasizes metaphorical rather than literal aspects of the metaphor term. For instance, for a metaphor such as "*reflect* concern," the context deemphasizes literal aspects of "reflect" related to light while emphasizing metaphorical aspects such as those related to communication (see Glucksberg, 2003, for discussion of dual reference in metaphor). Therefore, Shutova et al. applied the Van de Cruys et al. algorithm to paraphrase metaphor. To avoid selecting paraphrases that are also metaphorical (e.g., paraphrasing "*accelerate* change" as "*speed up* change"), Shutova et al. added an extra step in which selectional preference was used to further weight candidate paraphrases. The logic was similar to Fass's (1991) model mentioned earlier in that metaphor typically violates selectional preference. Therefore, candidate verbs that have higher selectional preference for the noun in the metaphor (and thus, are more likely to be literal) are weighted heavier in the final prediction. This step slightly improved the model as it helped filter out candidate paraphrases that were also metaphorical.

To assess the model, Shutova et al. (2012) had four volunteers judge the selected paraphrases. A paraphrase was counted correct if the top candidate from the list was judged correct by three of the four judges. Operationalized this way, Shutova et al.'s model was correct on 52% of paraphrases, which is fairly high considering the model is fully unsupervised.

Su, Huang, and Chen's recognition and interpretation model

Su et al. (2017) created a model that included both metaphor recognition and interpretation. Because metaphors involve topic and vehicle terms that are from dissimilar domains, the logic behind the recognition process was that the terms in a metaphor should be more dissimilar than terms in a literal statement. To calculate similarity, Su et al. (2017) used the Continuous-bag-of-words architecture to generate word vectors and used the cosine measure to index degree of similarity. They determined the threshold between metaphorical and literal by finding the cutoff point that maximized classification accuracy on a set of 40 literal and 40 metaphorical statements (a cosine of <.575 for Chinese and <.235 for English were the thresholds for metaphor classification). They then tested this threshold on a set of 250 new metaphor and 250 new literal statements (for both Chinese and English) and 85% of the statements were classified correctly in both languages.

For metaphor interpretation, Su et al. (2017) extracted properties of the vehicle by consulting databases that list properties of entities. These databases consist of properties extracted from mining text corpora however, and therefore are not manually coded inputs such as those used in Fass (1991) and Falkenhainer et al. (1989) described earlier. The relatedness between each of the vehicle's extracted properties and the topic word of the metaphor was computed. To gain a more accurate measure, the relatedness of the synonyms of each extracted property to the topic was also calculated. Su et al. (2017) were vague on how they calculated relatedness at this stage, but it is likely they used Continuous-bag-of-words vectors for the extracted features and computed cosine similarity to the topic (the same way they calculated relatedness in the recognition stage). Regardless, the property that has the highest relatedness with the topic is selected as the interpretation of the metaphor. The model then provides a literal paraphrase of the metaphor that places the extracted property in the paraphrase; for example, the metaphor *"inspiration is [a] spark"* was paraphrased as *inspiration is instantaneous*. To assess the model's interpretations, Su et al. had a group of five volunteers rate 100 metaphor paraphrases for both Chinese and English metaphors. The model performed considerably well as 87% and 85% of the paraphrases had average acceptability ratings above three (out of five) for Chinese and English, respectively.

As mentioned above, the paraphrase models of Shutova et al. (2012) and Su et al. (2017) are motivated by a different goal than the psychological models of Kintsch (2000), Utsumi (2011), and Terai and Nakagawa (2012). Shutova et al. and Su et al. come from the area of NLP in which the goal is to create systems that allow computers to decode human language. In contrast, Kintsch, Utsumi, and Terai and Nakagawa come from the area of cognitive science, and are therefore more interested in the cognitive processes involved in metaphor comprehension rather than generating accurate paraphrases. Nonetheless, paraphrase models are still relevant to understanding cognitive processes involved in metaphor. For instance, the fact that Shutova et al. found that selectional preference improved interpretation accuracy may suggest that humans also use selectional preference information to process metaphors. Similar to the argument that information retrieval parallels human semantic memory (Anderson & Milson, 1989; Jones, 1988), paraphrase models are likely most accurate when they "think" more similarly to a human. The major drawback with paraphrase models, however, is that they only select one interpretation rather than generating a new vector representing the complexity involved in metaphor meaning. Metaphors typically have multiple meanings, for instance, *"time is money"* could mean that time is valuable, limited, or can be wasted if not managed properly. In fact, Ortony (1975) has argued that the purpose of metaphor is to express an idea that cannot be expressed easily and concisely using literal language. Therefore, simply selecting one highly related property may be too narrow of an interpretation to capture the full meaning of the metaphor.

The major advantage of paraphrase models is that the interpretations they generate can be assessed directly. In contrast, the metaphor vectors generated by psychological models have to be compared to hand-selected neighbors. This obviously requires much interpretation on the part of the researcher as the vector itself usually cannot be interpreted directly.[3] Although these models provide insights into the psychological processes involved in metaphor comprehension, it would be interesting to see the actual interpretations these models generate based on the different processes involved. Furthermore, although the metaphor vectors are compared to human-generated interpretations, the interpretation tasks are quite simple, such as listing features (Terai & Nakagawa, 2012; Utsumi, 2011) or sentence completion with a literal substitute (Kintsch & Bowles, 2002) rather than an open response. Obviously this allows for a more quantitative evaluation, but these simple interpretation tasks may not fully capture the richness of the metaphor comprehension process.

Conclusions

Marr's (1982) tri-level analysis of computation and cognition is a useful way in which to characterize extant models of metaphor (see Dawson, 1998). The first, computational, level considers what the computational system does, such as the problems it must solve and why it even does what it does. With respect to metaphor, these would be questions on how do people understand the intended meaning of a sentence expressed nonliterally and the pragmatic conditions under which metaphor might be used. Virtually every current cognitive model works at this level. The second, algorithmic, level examines how the system does the types of things described in the computational level. At this

[3]Terai and Nakagawa's (2012) model is an exception to this. Because of the nature of their semantic space, the metaphor vector can be interpreted in terms to the probabilities of different features given the metaphor.

level, the representations of importance are identified and the step-by-step processes taken to "solve" the problems are quantified. Most extant theories take a stab at this but most do so metaphorically, rather than outline the steps with the exactitude necessary to construct a computer simulation of the envisioned processes. The vector space models described here are exceptions to our critique of current models; the representations are made explicit, how similarity is calculated is made clear and the various programs provide simulations that run without the need for human interpretation or intervention. That is not to say that other theoretical approaches will not be able to provide computational-real simulations, but until they do the theories have elements of a promissory note. The final, implementation, level described by Marr is the physical realization of the algorithms in neural systems, and although there are studies examining brain activation during the processing of metaphor and the like (see Ahrens et al., 2007; Eviatar & Just, 2006; Rapp, Leube, Erb, Grodd, & Kircher, 2004), to date no extant model of metaphor processing has approached this level, even the vector space models based on neural network computation.

The models described here have focused on metaphor processing from the Aristotelian perspective of the stretching of word meaning to provide a nonliteral metaphoric understanding. But the vector space approach is sufficiently flexible to be used to quantify other aspects of metaphor processing. For instance, Lemaire and Bianco (2003) found that LSA could model human processing time of referential metaphors (i.e., those in which the topic is introduced somewhere in the text, but the vehicle is not introduced until later after intervening text). Importantly, the words in the intervening context can influence processing time of the vehicle; if many words closely related to the metaphor appear in the context, it will facilitate processing. Lemaire and Bianco reasoned that the deeper into the semantic neighborhood of the vehicle one would have to search to find words related to the context, the longer the metaphor would take to process. Depth was defined as the number of neighbors searched before five words related to the context are found (the threshold for relatedness was a cosine $\geq .20$ between the target word vector and context vector). Lemaire and Bianco found that the depth of search required to find related words was closely related to human processing times. Moreover, while the examples and usage described here was for metaphors based on the stretching of word meaning, there is no reason the models could not be applied to the arousal of conceptual metaphors, such as seeing whether an expressed metaphor is best fit with a conceptual metaphor, or indeed whether some expressions better arouse the underlying metaphor than do other, ostensibly related, expressions. Indeed, Mohler et al. (2014) created a model for extracting conceptual metaphors from metaphorical expressions using an approach in which related terms are iteratively combined into "clusters." The clusters were then used to identify the source domains of different metaphorical expressions.

The value of these vector space models notwithstanding, such modeling is clearly in an early stage of development. There are obvious extensions already suggested in this tradition. For instance, we have discussed the basic differences between psychological and paraphrase models, but one can envision attempts to integrate the two. Paraphrase models could focus more on the psychological processes involved, and this may allow for more diverse paraphrases rather than selecting a single plausible paraphrase. In turn, psychological models should seek to have a clearly defined way to interpret the generated metaphor vectors. Also, perhaps combining both assessment strategies (comparison with human generated interpretations as well as direct ratings of the models' interpretations) would lead to a more complete assessment.

Although vector space models mostly focus on the similarity or distance between word meanings, recent work by Al-Azary and Buchanan (2017) examined how different types of semantic spaces influenced subjective ratings of metaphor comprehensibility, in particular by examining the metaphor topic and vehicle's semantic neighborhood density, a measure of how many closely related words there are to a target word in vector space. A word with many near neighbors (i.e., words with high cosines to the target word) is considered to reside in a high-density space; with few near neighbors the space is considered less dense. Al-Azary and Buchanan found that when both the topic and vehicle of novel metaphors came from low-density neighborhoods, the metaphors were rated as easier to comprehend. They theorized that with low-density neighborhoods, the relevant features of the metaphor are easier to find because there is less interference from close associates. Additionally, Katz and Al-Azary (2017) examined whether density plays a role in topic-vehicle asymmetries. One can expect further studies examining not only similarity but density, such as whether metaphors involving topics and vehicles from sparser semantic neighborhoods produce more emergent features than those from more dense neighborhoods. Finally, one can consider the role that space plays in other indices of metaphor in addition to recognition and interpretation, such as metaphor aptness, a focus of the non-computational models of both Tourangeau and Sternberg (1981) and Trick and Katz (1986), or measures of metaphor appreciation, such as perceived creativity and poeticality. For instance, Utsumi (2005) has speculated that greater semantic dissimilarity between topic and vehicle may illicit richer interpretations and greater poetic appreciation. Although Utsumi examined these hypotheses using subjective ratings of similarity, vector space representations may capture more fine-grained similarity, and perhaps a model of interpretation richness and poetic appreciation could be created using such representations.

In summary, vector space models have greatly improved the flexibility of computational models of metaphor. By providing a method for quantifying word similarity and locating words within a semantic space, metaphor models can now easily obtain vector representations of words relevant to different metaphors. Furthermore, as seen in Terai and Nakagawa (2012), Utsumi (2011), and Su et al. (2017), these methods can easily be applied in languages other than English (the former two in Japanese and the latter in Chinese). That being said, there are still major limitations, the most critical of which is that most of these models have only examined simple nominal metaphors (the exception being Shutova et al., 2012, who examined verb metaphors). Moreover, the models, encouraging as they are, have problems with complex and poetic metaphors (Kintsch, 2008). Although the scope of these models is limited currently to simple nominal and verb metaphors, they do handle those quite well, and the simulations have shed light on how humans process metaphors.

Acknowledgments

We thank Marc Joanisse, Ken McRae, and Paul Minda for their helpful suggestions.

Funding

This research was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant 06P0070.

References

Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic concepts of artificial neural network(ANN) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, *22*(5), 717–727.

Ahrens, K., Liu, H. L., Lee, C. Y., Gong, S. P., Fang, S. Y., & Hsu, Y. Y. (2007). FunctionalMRI of conventional and anomalous metaphors in Mandarin Chinese. *Brain and Language*, *100*(2), 163–171. doi:10.1016/j.bandl.2005.10.004

Al-Azary, H., & Buchanan, L. (2017). Novel metaphor comprehension: Semantic neighbourhooddensity interacts with concreteness. *Memory & Cognition*, *45*(2), 296–307.

Anderson, J. R., & Milson, R. (1989). Human memory: An adaptive perspective. *Psychological Review*, *96*(4), 703–719. doi:10.1037/0033-295X.96.4.703

Aristotle. (1952). Rhetoric. In W. D. Ross (Ed.); W. R. Roberts (Trans.), *The works of Aristotle Vol. 11: Theoretica, de rhetorica ad alexandrum, poetica* (pp. 1–51). Oxford, UK: Clarendon Press.

Aristotle. (1996). *Poetics*. (M. Heath, trans.). London, UK: Penguin Books.

Bowdle, B. F., & Gentner, D. (2005). The career of metaphor. *Psychological Review*, *112*(1), 193–216. doi:10.1037/0033-295X.112.1.193

Clark, S. (2015). Vector space models of lexical meaning. In S. Lappin & C. Fox (Eds.), *The Handbook of Contemporary Semantic Theory* (*2nd* ed., pp. 493–522). Malden, MA: Wiley-Blackwell.

Dawson, M. R. W. (1998). *Understanding cognitive science*. Oxford, UK: Blackwell Publishers.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407. doi:10.1002/(ISSN)1097-4571

Durda, K., & Buchanan, L. (2008). WINDSORS: Windsor improved norms of distance and similarity of representations of semantics. *Behavior Research Methods*, *40*(3), 705–712.

Eviatar, Z., & Just, M. A. (2006). Brain correlates of discourse processing: An fMRI investigation of irony and conventional metaphor comprehension. *Neuropsychologia*, *44*(12), 2348–2359. doi:10.1016/j.neuropsychologia.2006.05.007

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, *41*(1), 1–63. doi:10.1016/0004-3702(89)90077-5

Fass, D. (1991). met*: A method for discriminating metonymy and metaphor by computer. *Computational Linguistics*, *17*(1), 49–90.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*(2), 155–170. doi:10.1207/s15516709cog0702_3

Glucksberg, S. (2003). The psycholinguistics of metaphor. *Trends in Cognitive Sciences*, *7*(2), 92–96.

Glucksberg, S., & Keysar, B. (1990). Understanding metaphorical comparisons: Beyond similarity. *Psychological Review*, *97*(1), 3–18. doi:10.1037/0033-295X.97.1.3

Harman, D. (1986). An experimental study of the factors important in document ranking. In F. Rabitti (Ed.), *Association for Computing Machinery 9th Conference on Research and Development in Information Retrieval* (pp. 186–193). New York, NY: Association for Computing Machinery. doi:10.1145/253168.253206

Harris, Z. S. (1954). Distributional Structure. *WORD*, *10*(2–3), 146–162. doi:10.1080/00437956.1954.11659520

Hofmann, T. (1999). Probabilistic latent semantic analysis. In K. B. Laskey & H. Prade (Eds.), *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (pp. 289–296). San Francisco, CA: Morgan Kaufmann Publishers Inc. Retrieved from https://arxiv.org/ftp/arxiv/papers/1301/1301.6705.pdf

Jones, W. P. (1988). "As we may think"?: Psychological considerations in the design of a personal filing system. In R. Guindon (Ed.), *Cognitive Science and its Applications for Human-Computer Interaction* (pp. 235–287). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kameya, Y., & Sato, T. (2005). Computation of probabilistic relationship between concepts and their attributes using a statistical analysis of Japanese corpora. In *Proceedings of Symposium on Large-scale Knowledge Resources: LKR2005* (pp. 65–68), Tokyo, Japan. Retrieved from http://rjida.meijo-u.ac.jp/sato-www/reference/Kameya-LKR2005.pdf

Katz, A. N., & Al-Azary, H. (2017). Principles that promote bidirectionality in verbal metaphor. *Poetics Today*, *38*(1), 35–59. doi:10.1215/03335372-3716215

Kintsch, W. (2000). Metaphor comprehension: A computational theory. *Psychonomic Bulletin & Review*, *7*(2), 257–266.

Kintsch, W. (2008). How the mind computes the meaning of metaphor: A simulation based on LSA. In R. W. Gibbs Jr. (Ed.), *The Cambridge Handbook of Metaphor and Thought* (pp. 129–142). New York, NY: Cambridge University Press.

Kintsch, W., & Bowles, A. R. (2002). Metaphor comprehension: What makes a metaphordifficult to understand? *Metaphor and Symbol*, *17*(4), 249–262. doi:10.1207/S15327868MS1704_1

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by*. Chicago, IL: University of Chicago Press.

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. doi:10.1037/0033-295X.104.2.211

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25* (2–3), 259–284. doi:10.1080/01638539809545028

Lemaire, B., & Bianco, M. (2003). Contextual effects on metaphor comprehension: Experiment and simulation. In F. Detje, D. Dörner, & H. Schaub (Eds.), *Proceedings of the 5th International Conference on Cognitive Modeling (ICCM2003)* (pp. 153–158). Bamberg, Germany: Universitäts-Verlag Bamberg. Retrieved from http://cogprints.org/3205/1/iccm03_lemaire.pdf

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Proceedings of Advances in Neural Information Processing Systems* (Vol. *27*, pp. 2177–2185). Electronic proceedings, conference held in Montreal, Canada. Retrieved from http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization.pdf

Lin, D. (1998). Automatic retrieval and clustering of similar words. In P. Isabelle (Ed.), *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics* (pp. 768–774). Stroudsburg, PA: Association for Computational Linguistics. doi:10.3115/980691.980696

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co- occurrence. *Behavior Research Methods, Instruments, & Computers*, *28*(2), 203–208. doi:10.3758/BF03204766

Marr, D. (1982). *Vision: A computational investigation in the human representation of visual information*. San Francisco, CA: Freeman.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Proceedings of International Conference on Learning Representations Workshop*, Scottsdale, AZ. Retrieved from https://arxiv.org/abs/1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Proceedings of Advances in Neural Information Processing Systems* (Vol. 26, pp. 3111–3119). Electronic proceedings, conference held in Lake Tahoe, NV. Retrieved from https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

Mohler, M., Rink, B., Bracewell, D., & Tomlinson, M. (2014). A novel distributional approach to multilingual conceptual metaphor recognition. In J. Tsujii & J. Hajic (Eds.), *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 1752–1763). Dublin, Ireland: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/C14-1165

Ortony, A. (1975). Why metaphors are necessary and not just nice. *Educational Theory*, *25*(1), 45–53. doi:10.1111/edth.1975.25.issue-1

Ortony, A. (1979). Beyond literal similarity. *Psychological Review*, *86*(3), 161–180. doi:10.1037/0033-295X.86.3.161

Padó, S., & Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, *33*(2), 161–199. doi:10.1162/coli.2007.33.2.161

Rapp, A. M., Leube, D. T., Erb, M., Grodd, W., & Kircher, T. T. J. (2004). Neural correlates of metaphor processing. *Cognitive Brain Research*, *20*(3), 395–402. doi:10.1016/j.cogbrainres.2004.03.017

Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, *8*(10), 627–633. doi:10.1145/365628.365657

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back- propagating errors. *Nature*, *323*, 533–536. doi:10.1038/323533a0

Sahlgren, M. (2008). The distributional hypothesis. *Italian Journal of Linguistics*, *20*(1), 33–54.

Shutova, E., Van de Cruys, T., & Korhonen, A. (2012). Unsupervised metaphor paraphrasing using a vector space model. In *Proceedings of COLING 2012: Posters* (pp. 1121–1130), Mumbai, India. Retrieved from http://www.aclweb.org/anthology/C12-2109

Su, C., Huang, S., & Chen, Y. (2017). Automatic detection and interpretation of nominal metaphor based on the theory of meaning. *Neurocomputing*, *219*(5), 300–311. doi:10.1016/j.neucom.2016.09.030

Terai, A., & Nakagawa, M. (2012). A corpus-based computational model of metaphor understanding consisting of two processes. *Cognitive Systems Research*, *19*, 30–38. doi:10.1016/j.cogsys.2012.03.001

Tourangeau, R., & Sternberg, R. J. (1981). Aptness in metaphor. *Cognitive Psychology*, *13*(1), 27–55. doi:10.1016/0010-0285(81)90003-7

Touretzky, D. S., & Pomerleau, D. A. (1989). What's hidden in the hidden layers?. *Byte*, *14*(8), 227–233.

Trick, L., & Katz, A. N. (1986). The domain interaction approach to metaphor processing: Relating individual differences and metaphor characteristics. *Metaphor and Symbolic Activity*, *1*(3), 185–213. doi:10.1207/s15327868ms0103_3

Utsumi, A. (2005). The role of feature emergence in metaphor appreciation. *Metaphor and Symbol*, *20*(3), 151–172. doi:10.1207/s15327868ms2003_1

Utsumi, A. (2011). Computational exploration of metaphor comprehension processes using a semantic space model. *Cognitive Science*, *35*(2), 251–296. doi:10.1111/j.1551-6709.2010.01144.x

Van de Cruys, T., Poibeau, T., & Korhonen, A. (2011). Latent vector weighting for word meaning in context. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* (pp. 1476–1485). Stroudsburg, PA: Association for Computational Linguistics. Retrieved from https://hal.archives-ouvertes.fr/hal-00666475/document