# An Artificial Language Evaluation of Distributional Semantic Models

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

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

## Abstract

Recent studies of distributional semantic models have set up a competition between word embeddings obtained from predictive neural networks and word vectors obtained from count-based models. This paper is an attempt to reveal the underlying contribution of additional training data and post-processing steps on each type of model in word similarity and relatedness inference tasks. We do so by designing an artificial language, training a predictive and a count-based model on data sampled from this grammar, and evaluating the resulting word vectors in paradigmatic and syntagmatic tasks defined with respect to the grammar.

## 1 Introduction

The distributional tradition in linguistics (e.g., Harris, 1954) classically posits that a word's meaning can be estimated by its pattern of co-occurrence with other words. Modern distributional semantic models (DSMs) formalize this process to construct vector representations for word meaning from statistical regularities in large-scale corpora. A typical approach in NLP has been to apply dimensional reduction algorithms borrowed from linear algebra to a word-by-context frequency matrix representation of a text corpus (Deerwester et al. 1990, Landauer & Dumais, 1997). Words that frequently appear in similar contexts will have similar patterns across resulting latent components, even if they never directly co-occur (for reviews, see Jones, Willits, & Dennis, 2015; Turney & Pantel, 2010). These models dominated the literature over direct count methods for over two decades (Bullinaria & Levy, 2007, 2012). Recently, DSMs based on neural networks have rapidly grown in popularity (e.g., Bengio et al., 2003; Collobert et al., 2011; Mikolov et al., 2013). Given a word, the model attempts to predict the context words that it occurs with, or vice-versa. After training on a text corpus, the pattern of elements across the model's hidden layer come to reflect semantic similarities, i.e., will be similar for words that predict similar contexts even if those words do not predict each other. In this sense, neural embedding models come to a distributed vector representation of word meaning that is reminiscent of traditional dimensional reduction DSMs, albeit with a considerably different learning algorithm.

Mikolov et al. (2013a, 2013b) have demonstrated state-of-the-art performance using a neural embedding model with an efficient objective function called `word2vec`. This model rapidly emerged as the leader of the DSM pack, outperforming other models on a broad range of lexical semantic tasks (Baroni et al. 2014). However, since the early surge in excitement for `word2vec`, the literature has now become more focused on trying to understand the conditions under which embedding or traditional DSMs are optimal. Levy and Goldberg (2014) demonstrated analytically that `word2vec` is implicitly factorizing a word-by-context matrix whose cell values are shifted PMI values. In other words, the objective function and the input to `word2vec` are formally equivalent to traditional DSMs; thus the models should behave alike in the limit. The distinction is really one of process and parameterization. With optimum parameterization of traditional DSMs, more recent research is finding insignificant performance differences between `word2vec` and SVD factorizations of a PMI matrix (Sahlgren & Lenci, 2016). Levy et al. (2015) even found a slight advantage for a factorization of the bias shifted log-count matrix and for traditional PPMI over `word2vec` on some tasks when hyperparameters were optimized.

One general distinction between the two types of models is that neural embedding models such as `word2vec` seem to underperform when the training corpus is small, particularly for low-frequency words (Asr et al., 2016; Sahlgren &

*Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 134–142,
Vancouver, Canada, August 3 - August 4, 2017. ©2017 Association for Computational Linguistics

Lenci, 2016). Levy et al. (2015) note that there is often a benefit in word2vec of tuning a larger parameter space over using a larger training corpus. With limited-data mining scenarios becoming more common, a better understanding of how model type and corpus size interact with optimal parameterization is an important topic of inquiry.

Secondly, interest has shifted from trying to determine the best overall model towards a better understanding of what kinds of word relations each model is best at learning, and under what parameterizations. Count-based PMI models are very good at representing first-order statistical patterns that reflect **syntagmatic** relationships in language (aka “relatedness” data). In contrast, the training scheme used by word2vec attempts to optimize it for detecting second-order statistical patterns that reflect **paradigmatic** relationships in language (aka “similarity” data). Indeed, this was the pattern demonstrated by Levy et al. (2015): After tuning hyperparameters, word2vec performed best on similarity-based tasks while PPMI performed best on relatedness tasks. SVD-based models attempt to represent both statistical patterns. This count-based model outperformed both word2vec and PPMI in Levy et al. on both types of relations when standard parameter sets were used; however, the advantage disappeared when hyperparameters were tuned. Standard word2vec is optimized for paradigmatic tasks but architectural adaptations exist to make the model better suited for syntagmatic tasks (e.g., Kiela et al., 2015; Ling et al., 2015). Making a model better at one type of task might come at the cost of making it worse at the other if the two types of word relations are orthogonal (Andreas & Klein, 2014; Mitchell & Steedman, 2015). Optimizing for a particular task is also closely tied to the issue of training data size (Melamud et al., 2016).

Finally, both of these issues are intricately tied to post-processing of the embeddings. Levy et al. (2015) inspired by Pennington et al., (2014) pointed out an important parametrization of the word2vec model, where co-occurrence information encoded between hidden and output layers (context vectors) are used as well as weighs between the input and hidden layers (word vectors) to construct the final word embeddings (w+c representation). When calculating word similarity based on this composite representation, a mixture between first- and second-order coocurrence information are considered. This is remarkably similar to cognitive models that construct composite memory representations from both paradigmatic and syntagmatic information (Jones & Mewhort, 2007). Recent empirical studies in developmental psychology have found that children learn word relations that have both sources of information before relations with either source alone (Unger et al., 2016). Levy et al. (2015) found a consistent benefit for word2vec and PPMI when the *w+c* post-processing combination was applied. Even though, this is an efficient adaptation in that the scheme does not require retraining, most studies on word similarity and relatedness have only employed the default word2vec setting (i.e., only using word vectors) and the usefulness of context vectors has been left underexplored.

It is very plausible to assume that the above three issues (corpus size, relation type, post-processing) interact: Higher-order paradigmatic word relations likely require more training data to discover, and the merging of *w+c* blends different relation types. The goal of this paper is to elaborate on the effect of corpus size and post-processing on the reflection of syntagmatic and paradigmatic relations between words within the resulting vector space. It has proven impossible in psycholinguistics to select real words that cleanly separate paradigmatic and syntagmatic relations (McNamara, 2005). Hence, we opted to bring the statistical structure of the language under experimental control using an artificial language adapted from Elman (1990). Unlike in natural language corpora, the sources are independent: e.g., *dog* never directly appears with *cat*, and hence any learned relation between them could not be due to first-order information. Thus by defining crisp semantic categories and sentence frames, we investigate how first and second-order co-occurrence information sources are consumed and represented in terms of similarity between words by *count-based* and *predictive* DSMs. Given current uncertainty in the literature on the role of corpus size, relation type, and *w+c* post-processing regarding the performance of various DSM architectures, this approach affords experimental control to evaluate relative performance as a factorial combination of information sources and parameters while controlling for the many confounding factors that exist in natural language corpora; including the ambiguity of similarity vs. relatedness of two words in evaluation datasets. Section 2 describes our framework in details, and section 3 presents several experiments exploring the capacity of count vs. predict DSMs in modeling relations between words.

## 2 Experiment Setup

### 2.1 Creation of Corpus

The artificial language grammar that we use for generating sentences in our test corpora is depicted in Table 1. This grammar was first introduced by Elman (1990) in his exploration of language modeling by Recurrent Neural Networks (RNNs). The language consists of a small vocabulary, a set of explicitly defined semantic categories on top of the vocabulary, and finally, a set of syntactic rules or possible sentence frames, which specifies how words can be put together in a sentence with regard to their semantic categories. The language generation algorithm enumerates all possible sentences in the language and the corpus generator returns a random sample of the language using a uniform distribution across sentence types. The corpus size is a variable in our experiments, and we mention explicitly when we repeat an experiment by re-sampling a corpus to validate the results on the semantic similarity tasks.

### 2.2 Semantic Similarity Tasks

All experiments in the current paper are centered on the idea that, at least, two types of semantic similarity can be identified for word pairs.

Table 1. Artificial language grammar (Elman 1990)

| Sentence Frames | Example |
|---|---|
| NOUN-HUM VERB-EAT NOUN-FOOD | *man eat cookie* |
| NOUN-HUM VERB-PERCEPT NOUN-INANIM | *woman see book* |
| NOUN-HUM VERB-DESTROY NOUN-FRAG | *man smash glass* |
| NOUN-HUM VERB-INTRAN | *woman sleep* |
| NOUN-HUM VERB-TRAN NOUN-HUM | *man chase woman* |
| NOUN-HUM VERB-AGPAT NOUN-INANIM | *woman brake book* |
| NOUN-HUM VERB-AGPAT | *man move* |
| NOUN-ANIM VERB-EAT NOUN-FOOD | *cat eat cookie* |
| NOUN-ANIM VERB-TRAN NOUN-ANIM | *mouse see cat* |
| NOUN-ANIM VERB-AGPAT NOUN-INANIM | *cat chase mouse* |
| NOUN-ANIM VERB-AGPAT | *mouse move* |
| NOUN-INANIM VERB-AGPAT | *rock move* |
| NOUN-AGRESS VERB-DESTROY NOUN-FRAG | *dragon brake plate* |
| NOUN-AGRESS VERB-EAT NOUN-HUM | *monster eat man* |
| NOUN-AGRESS VERB-EAT NOUN-ANIM | *dragon eat cat* |
| NOUN-AGRESS VERB-EAT NOUN-FOOD | *monster eat cookie* |
| Semantic Categories | |
| NOUN-HUM: [man, woman] | |
| NOUN-ANIM: [cat, mouse] | |
| NOUN-AGRESS: [dragon, monster] | |
| NOUN-INANIM: [book, rock] | |
| NOUN-FRAG: [glass, plate] | |
| NOUN-FOOD: [cookie, sandwich] | |
| VERB-INTRAN: [think, sleep] | |
| VERB-TRAN: [see, chase] | |
| VERB-PERCEPT: [smell, see] | |
| VERB-AGPAT: [move, break] | |
| VERB-DESTROY: [break, smash] | |
| VERB-EAT: [eat] | |

Thus, we define two distinct methods to evaluate performance of the DSMs in learning semantic similarity from our artificial language—the syntagmatic task and the paradigmatic task.

**Syntagmatic task:** the objective of this task is to identify word pairs that can occur in context together (here the scope of a sentence). For example, the word pair *smash* and *cookie* cannot appear in each other's context according to the grammar in Table 1, because no legal sentence frame includes the semantic category of both words. Conversely, the word pair *eat* and *cookie*s are related in the sense that the two words can co-occur within a sentence. Evaluation of the vectors produced by different DSMs in this task is based on the cosine similarity between words occurring in common vs. different context frames and is calculated by the following accuracy measure:

$$Accuracy_{syn} = Avg\ sim(w_i, w_j) \\ - Avg\ sim(w_k, w_l)$$

where ($w_i$ , $w_j$) is indicative of the word pairs in the vocabulary that appear together in at least one sentence frame, and ($w_k$ , $w_l$) is indicative of word pairs that do not appear in any common frame given their semantic categories (e.g., *glass* and *chase* belong to *NOUN-FRAG* and *VERB-TRANS*, respectively, which never co-occur within a sentence).

The syntagmatic task is a strict version of finding first-order related, directly co-occurring, or similar topic words in a natural language. Since word pairs are exclusively labeled as co-occurring vs. non-co-occurring based on the grammar of the artificial language, we will have the possibility to look into the performance of the DSM models in drawing syntagmatic similarities without having to deal with other confounds present in natural languages. This type of evaluation is almost impossible in a natural language given the openness of the semantic categories and enormous grammar size. In our modeling framework, if words are distributed in a DSM mostly based on first-order co-occurrence information, accuracy of the syntagmatic task would be high.

**Paradigmatic task:** two words should be similar if they tend to occur in similar contexts even if they never co-occur in the same sentence. Our paradigmatic task is defined based on this intuition, and the idea of taxonomically similar words in natural languages. According to Table

1, if two words come from the same semantic category (e.g., *man* and *woman*) they appear in similar sentence frames, thus ideally (when all possible sentence formulations exist in the generated sample of the language) they should be found as fully substitutable words. The paradigmatic task evaluates the quality of word vectors generated by a DSM by calculating the cosine similarity of word pairs belonging to same vs. different sematic categories.

$$Accuracy_{par} = Avg\ sim(w_i, w_j) - Avg\ sim(w_k, w_l)$$

where ($w_i$ , $w_j$) indicates all word pairs coming from same semantic categories, and ($w_k$ , $w_l$) indicates word pairs belong to different semantic categories. Based on this formulation, the paradigmatic accuracy of a model emphasizing second-order information would be higher than a model favoring first-order information to distribute words in the vector space. The reason is that, in the former model, the cosine similarity between vectors of interchangeable words like *man* and *woman* would converge to 1, or will be at least higher than similarity between other word vectors.[1] Both $Accuracy_{syn}$ and $Accuracy_{par}$ are bounded measures within the range of [-2, 2]; in practice though, they tend to come out within the range of [0, 1].

The above two tasks define the basics of our discriminative approach to investigate which models or parameter settings work best for each type of semantic similarity induction.

### 2.3 Distributional Methods

In our experiments, we use the implementations of `word2vec` Skip-Gram with Negative Sampling (SGNS) and PMI matrix factorization via Singular Value Decomposition (SVD) by Levy et al. (2015).

The Skip-gram model (SGNS) is one of the two `word2vec` architectures that predicts based on a target word one of its context words at a time. Error of prediction is calculated in the output via softmax and back-propagated to update two weight matrices: the context matrix (*CM*) between the output and the hidden layer $[]_{vd}$, and the word matrix (*WM*) between the input and the hidden layer $[]_{vd}$ ,where $v$ is the vocabulary size and $d$ is the size of the hidden layer, thus dimensionality of the final word vectors. In the majority of previous work, the word matrix was used as the final output of the model. When context words are sampled from the same vocabulary as that of target words, the final *CM* will have the same dimensionality as *WM*, thus it can also be used as a semantic representation of the words. Averaging both matrices for a final word representation, rather than just the *WM,* is an optional post-processing method indicated by $w+c$.

Singular Value Decomposition (SVD) is a classic representation learning technique for projecting data into a new, and usually, smaller feature space. Other similar techniques in machine learning include eigenvalue decomposition, the basis of Principle Component Analysis. The SVD model in our study is representative of the count-based distributional semantic models. It begins by calculating a $v*v$ matrix of point-wise mutual information between word-context pairs. The matrix is then factorized and reduced to a $v*d$ matrix, where each row will be a word vector in the new semantic space.

### 2.4 Implementation and Parameter Balancing

In all our experiments, we try to equate the two models by keeping the common parameters constant and iterating over different values of the method-specific parameters to obtain the best performance for each.

**Fixed parameters:** parameters that we keep constant throughout all experimental conditions are the context window size (set to 2, in order to cover all words within a sentence in the artificial grammar), subsampling & dynamic context (set to off; no frequency-based smoothing or prioritization is applied to co-occurrence counts), rare word removal (set to off, no minimum cut-off is applied to context words). Therefore, in all experimental conditions that result from manipulating other parameters exactly the same word-context population is extracted from a given corpus and fed as input data to the SGNS and SVD models. We also use one iteration (epoch) in SGNS to keep it equated with SVD, and examine the effect of re-occurrences by manipulating the corpus size instead.

**Variable parameters:** for comparative experiments on small vs. big data, we generate 5

[1] The paradigmatic task can also be defined based on higher-level taxonomic relations. For example, given the grammar in Table 1, we expect models to cluster Verbs and Nouns because each of these higher-level word types share some within-category contextual similarities and between-category differences (e.g., all nouns in the grammar have a verb in context, whereas verbs don't have verbs in their context). In section 3.5 where semantic spaces are visualized we will return to this important point, but for the rest of our experiments model performance is evaluated based on the two basic tasks defined above.

independent corpora of each size (between 1K and 30K sentences) according to the sampling procedure described in Section 2.1. There are three important parameters that strongly affect the performance of the models, but since they are not the focus of our study we chose their values through a performance maximization procedure in all our experiments. One parameter called *dim* is the number of reduced dimensions or the size of final vectors, which is enumerated between 2 and 14 in our experiments. The other parameter *neg* is only applicable to SGNS and indicates the number of negative samples (we try between zero and 6 negative samples). Finally, a parameter in SVD determines the asymmetry of factorization, which was simulated with 0, 0.5 and 1 *eig* (for more details refer to Levy et al., 2015).

## 3 Results

### 3.1 Vanilla Comparison

Our first comparison explores the overall performance of the two DSMs with their common post-processing practice. We only use the W matrix to construct the word vectors after training SGNS, and the SVD factorization is also performed in its default manner. As explained in 2.4, we sampled five corpora of each size and measured the maximum likelihood of a model's performance by manipulating the variable parameters.

Table 2 shows that both models had very low overall accuracies in grouping syntagmatically related words. This observation indicates that, by default, both SVD and SGNS consume first-order co-occurrence information but infer second-order information, i.e., paradigmatic similarities between words by generalizing over context types in which two words can be seen. This finding suggests that neither of the models with its default configuration is suitable for performing word relatedness tasks. Reported best performances in the table for SVD were obtained at eig = 0.0, and for SGNS at neg = 1. Optimal dimensionality was variable but always above 5.

Table 2. Vanilla setup accuracy in paradigmatic and syntagmatic tasks with different size training corpuses.

| Corpus size | Method | Paradigmatic | Syntagmatic |
|---|---|---|---|
| 1K | SVD | **0.828** | **0.253** |
| | SGNS | 0.535 | 0.113 |
| 10K | SVD | **0.832** | **0.258** |
| | SGNS | 0.775 | 0.092 |

### 3.2 Corpus Size

Accuracy scores in Table 2 suggest that, even with small training data SVD can produce good vectors for the paradigmatic task. However, the performance of SGNS increases with more training data. This quick observation is consistent with previous findings regarding the superior performance of count models on word similarity and categorization tasks when models were trained on small corpora and with their default post-processing setting (Asr et al., 2016; Sahlgren & Lenci, 2016). The main reason stated in the literature is that SGNS requires tuning a large number of parameters and seeing more and more data (either through extra epochs or by feeding in a larger corpus of the same distribution of words and sentences) helps the model to converge. In the next sections we will see how otherwise we could enhance this model's performance, possibly in both syntagmatic and paradigmatic tasks.

### 3.3 Inclusion of Context Vectors

We hypothesized that using a post-processing setup emphasizing first-order information should enhance models' performance in the syntagmatic task. To test this, we repeated experiments on training corpora of size 1K to 30K with the alternative post-processing approaches (inclusion of context vectors, i.e., *w+c* vs. *w,* which was the default setting).

Figure 1 shows that the inclusion of context vectors enhances the accuracy of both models in the syntagmatic task (red lines are on top of the blue lines). This enhancement is more pronounced in the SGNS model: more data increases the accuracy of syntagmatic similarity inference consistently when the *w+c* option is used. SVD also benefits from a *w+c* equivalent setting proposed by Levy & Goldberg (2015) in performing the syntagmatic task, however the enhancement is tightly bounded for this model.

For the paradigmatic task, we expected an inverse pattern: explicit inclusion of first-order co-occurrence information in similarity measurement by considering both word and context vectors should hurt model's performance because only second-order information is important for the paradigmatic task. We can see in Figure 2 that our hypothesis is supported for SVD, where the accuracy declines significantly with the inclusion of the context vectors (compare the red and blue dotted lines). However, the SGNS model does not exhibit a dramatic change of performance in the paradigmatic task with or without the *w+c* option (compare the solid lines). In fact, the performance in the paradigmatic task was slightly enhanced too. Putting this together with what we saw above regarding SGNS performance in the syntagmatic task brings us to an interesting conclusion about the "optimal parameter setting" for this model: using the *w+c* option is a good choice adding to the robustness of SGNS, particularly when unsure of which type of similarity inference we would like the model to perform at the end. The SVD model, on the other hand, does not show the capability to learn both tasks at the same time; it gets better in one at the expense of the other. In the next section we try to explain this difference by looking into the way the two models distribute words within the high dimensional vector space.

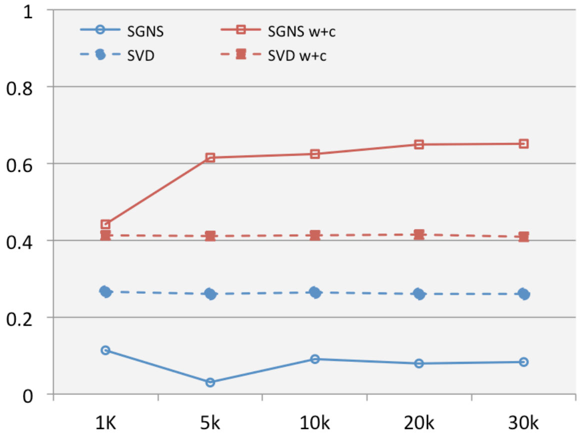

Figure 1. Accuracy of SGNS and SVD with word only vs. word+context vectors trained on corpuses of different sizes (1K to 30K sentences) in the syntagmatic task.

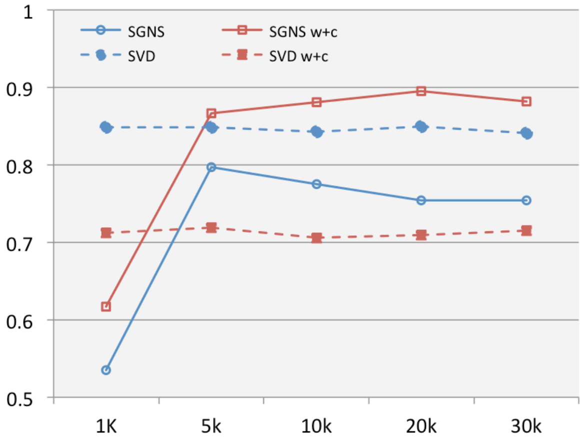

Figure 2. Accuracy of SGNS and SVD with word only vs. word+context vectors trained on corpuses of different sizes (1K to 30K sentences) in the paradigmatic task.

### 3.4 Metric Space Expansion/Compression

The above experiments showed a lower ceiling for SVD performance compared to SGNS in both tasks when sufficient data was available to the models and the parameter space was thoroughly explored. In order to explain this observation, we took a closer look at the vectors generated by each model and specifically examined the range of the similarity scores of all word pairs in the vocabulary. We found that SVD generated numerically closer vectors compared to SGNS. This results in a smaller range of similarity scores: totally interchangeable words, such as *man* and *woman* get a cosine similarity score close to 1.0; completely different words (that neither appear in a sentence together, nor share similar contexts) such as *glass* and *chase* get a negative similarity score typically close to 0.0, or around -0.5 in a best case scenario.

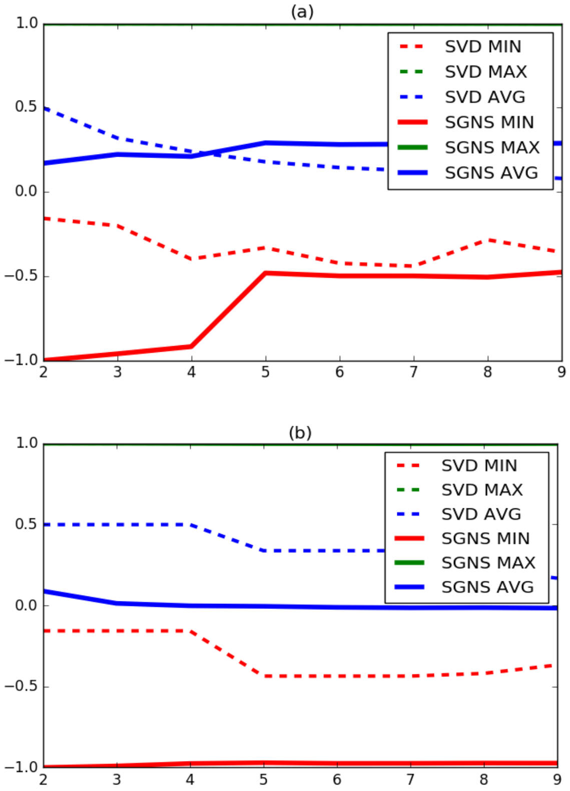

Figure 3. Spectrum of similarity scores between words in SVD and SGNS (10K corpus, neg = 1, eig = 0, dim = 2 to 9 on the x-axis): (a) with *w* and, (b) with *w+c* post-processing.

Figure 3 depicts the minimum, maximum and average similarity scores obtained for all word pairs from the vocabulary through repeated experiments on a 10K corpus by manipulating the dimensionality (x-axis). It is almost the same for SGNS and SVD when the word-only post-processing is applied, but as soon as the context vectors are included, the spectrum of similarity scores widens up for SGNS. This investigation may explain why SVD is unable to manifest paradigmatic and syntagmatic relations at the same time.

SVD does not get a huge benefit from more training data or the post-processing step for inclusion of the context vectors. The underlying reason is that SVD always uses a sub-space of the entire similarity spectrum [-0.5, 1.0] so everything is squeezed – we refer to this phenomenon as *space compression*, which we hypothesize is due to the limitations of the dimensionality reduction mechanism. On the other hand, the distribution of words in the vector space obtained from SGNS changes drastically both by training on more data and considering context vectors.

As Figure 3 shows, SGNS has the capacity to use up the entire similarity spectrum [-1.0, 1.0], i.e., *space expansion*. We conjecture that this is due both to the design of the objective function and to the larger number of parameters in the neural model being updated independently, making it a more flexible method to encode fine-grained differences between word groups, while keeping them in meaningful clusters. More data helps the model fine-tune its parameters. Furthermore, averaging the word and context vectors provides an ensemble voting for syntagmatic (relatedness) and paradigmatic (similarity) at the same time.

### 3.5 Word Clusters in the Semantic Space

The space expansion of the SGNS model by inclusion of the context vectors can be visualized with a 2-dimensional projection of the vectors obtained from *w* vs. *w+c* post-processing conditions, depicted in Figures 4 and 5 respectively. A comparison between the two plots shows how the vicinity of paradigmatically similar words (interchangeable words such as *cat* and *mouse*) can be preserved while syntagmatic clusters are emphasized (*cat* and *chase*) by inclusion of context vectors.

It is important, however, to note that higher-level paradigmatic relations are negatively affected as the model tries to bring syntagmatically related words closer to one another. For example, verbs and nouns (clustered in gray ovals in Figures 4), which are paradigmatically different, get mixed up once the syntagmatic clusters start to shape (gray rectangles in Figure 5). On the other hand, nouns referring to animate categories (that have some level of paradigmatic similarity) fall apart in the *w+c* space (red dashed cluster in Figures 4, distorted in Figure 5). These observations emphasize the importance of the post-processing choices based on the final inferences we expect from the model. When generalized to a natural language setting, the models depending on the w+c parameterization would demonstrate synonymy, similarity and associative relatedness differently.

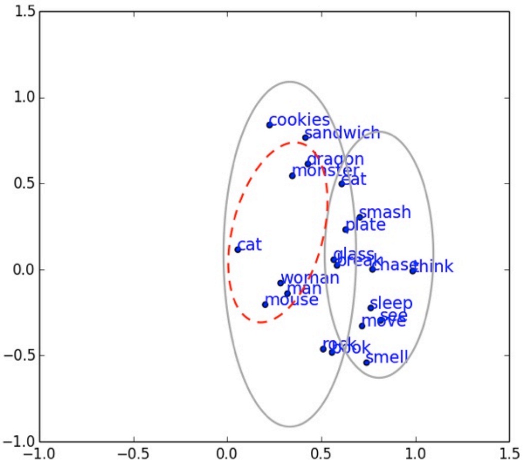

Figure 4. Paradigmatic clusters in SGNS *w* vector space; Syntagmatic clusters not easily identified (10K corpus, dim = 14, neg = 1)

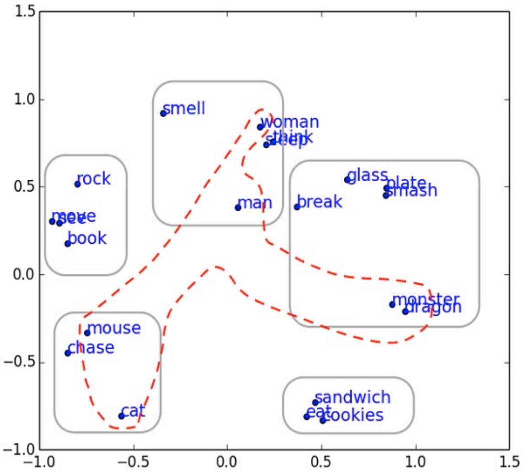

Figure 5. Clear syntagmatic clusters in SGNS *w+c* vector space; some paradigmatically related words are kept together and some have fallen apart (10K corpus, dim = 14, neg = 1)

One should consider that while dimensionality reduction to two dimensions is possible and helpful for visualization purposes, these images do not reflect the exact distances between words in the high-dimension space. Therefore, these observations should be understood in combination with other results, e.g., similarity spectrums demonstrated in the previous section.

## 4 Conclusion

We proposed a methodology based on artificial language generation for studying distributional semantic models. This methodology was inspired by the prominent study of Elman (1990) and we mainly selected that to bring confound factors in natural languages under control while assessing the effect of model parameters on produced word vectors.

The experiments in this paper revealed an interaction between the training corpus size and a variety of parameter settings of two opponent DSMs in word similarity/relatedness evaluation. Confirming previous findings with small training data, we showed that SVD could easily organize words based on paradigmatic similarities obtained from second-order co-occurrence information, whereas SGNS needed more data to acquire the same type of knowledge. When it comes to syntagmatic relatedness between words, both models required accurate parameter settings. In particular, the default configuration of both SVD and SGNS aims at optimizing the space in a way that paradigmatically similar words are put together.

The optimal setting of the SGNS for an overall superior performance in both paradigmatic and syntagmatic tasks involved the inclusion of context vectors, which is not the typically tested setting of `word2vec` in previous studies. Our analysis of similarity scores between vectors generated for all words in the artificial language showed that averaging word and context vectors would result in a more organized SGNS vector space. The equivalent post-processing of the matrices in SVD for explicit inclusion of first-order similarity suggested by Levy et al. (2015) enhanced the performance of this model in the syntagmatic (relatedness) task only in the expense of making it worse for the paradigmatic (similarity) task.

Our observations suggest that SVD has some limitations in populating the distributional space as evenly as SGNS; thus it always comes up with vectors that are on average closer to one another. Further study is needed to explain this finding in a fundamental way perhaps via mathematical derivations. The trade-off between performance in paradigmatic and syntagmatic task, specially for the SVD model, can explain the occasional superiority and inferiority of this model against the neural opponents in previous studies: similarity and relatedness rankings for words in natural languages manifest a mixture of paradigmatic and syntagmatic relations among words, thus a certain SVD model (with its post-processing optimized for reflecting either type of relation) might outperform SGNS in one task and not in the other.

Our experiments were a first step towards understanding the differences between classic and neural distributional models in a more controlled setting. The proposed methodology can be used in future research, e.g. to assess the effect of vocabulary and grammar size on resulting word vectors by different models, and in turn to select the right distributional approach in specific research context. We hope also that our work will initiate a general methodology for understanding the mechanism of neural networks employed in a variety of natural language processing tasks.

## Acknowledgement

We are thankful to our reviewers for their helpful feedback on the initial version of the paper and suggestions for extension of the work. This research was funded by grant R305A140382 from the Institute of Education Sciences.

## References

Andreas, J., & Klein, D. (2014). How much do word embeddings encode about syntax? In *Proceedings of ACL* (pp. 822-827).

Asr, F. T., Willits, J. A., & Jones, M. N. (2016). Comparing Predictive and Co-occurrence Based Models of Lexical Semantics Trained on Child-directed Speech. In *Proceedings of the Annual Meeting of Cognitive Science Society.*

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

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Machine Learning Research*, *3*(Feb), 1137-1155.

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

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493-2537.

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.

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

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

Kiela, D., Hill, F., & Clark, S. (2015). Specializing word embeddings for similarity or relatedness. In *Proceedings of EMNLP*.

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

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

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2), 211.

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

Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, *3*, 211-225.

Li, J., Chen, X., Hovy, E. and Jurafsky, D. (2016). Visualizing and Understanding Neural Models in NLP. In *Proceedings of NAACL.*

Ling, W., Dyer, C., Black, A., & Trancoso, I. (2015). Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of ACL-HLT* (pp. 1299-1304).

McNamara, T. P. (2005). Semantic priming: Perspectives from memory and word recognition. *Psychology Press.*

Melamud, O., McClosky, D., Patwardhan, S., & Bansal, M. (2016). The role of context types and dimensionality in learning word embeddings. *arXiv preprint arXiv:1601.00893.*

*Miller, G. A. (1958). Free recall of redundant strings of letters. Journal of Experimental Psychology, 56(6), 485.*

Mitchell, J., & Steedman, M. (2015). Orthogonality of syntax and semantics within distributional spaces. In *Proceedings of ACL.*

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., and Dean., J. (2013b) Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS*, 2013.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global Vectors for Word Representation. In *EMNLP* (Vol. 14, pp. 1532-43).

Sahlgren, M., & Lenci, A. (2016). The Effects of Data Size and Frequency Range on Distributional Semantic Models. *arXiv preprint arXiv:1609.08293*.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1), 141-188.

Unger, L., Fisher, A. V., Nugent, R., Ventura, S. L., & MacLellan, C. J. (2016). Developmental changes in semantic knowledge organization. *Journal of Experimental Child Psychology*, *146*, 202-222.