A Compositional and Interpretable Semantic Space

Alona Fyshe,[1] **Leila Wehbe,**[1] **Partha Talukdar,**[2] **Brian Murphy,**[3] **and Tom Mitchell**[1]
[1] Machine Learning Department, Carnegie Mellon University, Pittsburgh, USA
[2] Indian Institute of Science, Bangalore, India
[3] Queen's University Belfast, Belfast, Northern Ireland
afyshe@cs.cmu.edu, lwehbe@cs.cmu.edu, ppt@serc.iisc.in,
brian.murphy@qub.ac.uk, tom.mitchell@cs.cmu.edu

Abstract

Vector Space Models (VSMs) of Semantics are useful tools for exploring the semantics of single words, and the composition of words to make phrasal meaning. While many methods can estimate the meaning (i.e. vector) of a phrase, few do so in an interpretable way. We introduce a new method (CNNSE) that allows word and phrase vectors to adapt to the notion of composition. Our method learns a VSM that is both tailored to support a chosen semantic composition operation, and whose resulting features have an intuitive interpretation. Interpretability allows for the exploration of phrasal semantics, which we leverage to analyze performance on a behavioral task.

1 Introduction

Vector Space Models (VSMs) are models of word semantics typically built with word usage statistics derived from corpora. VSMs have been shown to closely match human judgements of semantics (for an overview see Sahlgren (2006), Chapter 5), and can be used to study semantic composition (Mitchell and Lapata, 2010; Baroni and Zamparelli, 2010; Socher et al., 2012; Turney, 2012).

Composition has been explored with different types of composition functions (Mitchell and Lapata, 2010; Mikolov et al., 2013; Dinu et al., 2013) including higher order functions (such as matrices) (Baroni and Zamparelli, 2010), and some have considered which corpus-derived information is most useful for semantic composition (Turney, 2012; Fyshe et al., 2013). Still, many VSMs act like a black box - it is unclear what VSM dimensions represent (save for broad classes of corpus statistic types) and what the application of a composition function to those dimensions entails. Neural network (NN) models are becoming increasingly popular (Socher et al., 2012; Hashimoto et al., 2014; Mikolov et al., 2013; Pennington et al., 2014), and some model introspection has been attempted: Levy and Goldberg (2014) examined connections between layers, Mikolov et al. (2013) and Pennington et al. (2014) explored how shifts in VSM space encodes semantic relationships. Still, interpreting NN VSM dimensions, or factors, remains elusive.

This paper introduces a new method, Compositional Non-negative Sparse Embedding (CNNSE). In contrast to many other VSMs, our method learns an *interpretable* VSM that is tailored to suit the semantic composition function. Such interpretability allows for deeper exploration of semantic composition than previously possible. We will begin with an overview of the CNNSE algorithm, and follow with empirical results which show that CNNSE produces:

1. more interpretable dimensions than the typical VSM,
2. composed representations that outperform previous methods on a phrase similarity task.

Compared to methods that do not consider composition when learning embeddings, CNNSE produces:

1. better approximations of phrasal semantics,
2. phrasal representations with dimensions that more closely match phrase meaning.

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 32–41,
Denver, Colorado, May 31 – June 5, 2015.

2 Method

Typically, word usage statistics used to create a VSM form a sparse matrix with many columns, too unwieldy to be practical. Thus, most models use some form of dimensionality reduction to compress the full matrix. For example, Latent Semantic Analysis (LSA) (Deerwester et al., 1990) uses Singular Value Decomposition (SVD) to create a compact VSM. SVD often produces matrices where, for the vast majority of the dimensions, it is difficult to interpret what a high or low score entails for the semantics of a given word. In addition, the SVD factorization does not take into account the phrasal relationships between the input words.

2.1 Non-negative Sparse Embeddings

Our method is inspired by Non-negative Sparse Embeddings (NNSEs) (Murphy et al., 2012). NNSE promotes interpretability by including sparsity and non-negativity constraints into a matrix factorization algorithm. The result is a VSM with extremely coherent dimensions, as quantified by a behavioral task (Murphy et al., 2012). The output of NNSE is a matrix with rows corresponding to words and columns corresponding to latent dimensions.

To interpret a particular latent dimension, we can examine the words with the highest numerical values in that dimension (i.e. identify rows with the highest values for a particular column). Though the representations in Table 1 were created with our new method, CNNSE, we will use them to illustrate the interpretability of both NNSE and CNNSE, as the form of the learned representations is similar. One of the dimensions in Table 1 has top scoring words *guidance, advice* and *assistance* - words related to help and support. We will refer to these word list summaries as the dimension's **interpretable summarization**. To interpret the meaning of a particular word, we can select its highest scoring dimensions (i.e. choose columns with maximum values for a particular row). For example, the interpretable summarizations for the top scoring dimensions of the word *military* include both positions in the military (e.g. commandos), and military groups (e.g. paramilitary). More examples in Supplementary Material (`http://www.cs.cmu.edu/~fmri/papers/naacl2015/`).

NNSE is an algorithm which seeks a lower dimensional representation for w words using the c-dimensional corpus statistics in a matrix $X \in \mathbb{R}^{w \times c}$. The solution is two matrices: $A \in \mathbb{R}^{w \times \ell}$ that is sparse, non-negative, and represents word semantics in an ℓ-dimensional latent space, and $D \in \mathbb{R}^{\ell \times c}$: the encoding of corpus statistics in the latent space. NNSE minimizes the following objective:

$$\underset{A,D}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{w} \left\| X_{i,:} - A_{i,:} \times D \right\|^2 + \lambda_1 \left\| A_{i,:} \right\|_1 \tag{1}$$

$$\text{st: } D_{i,:} D_{i,:}^T \leq 1, \forall\, 1 \leq i \leq \ell \tag{2}$$

$$A_{i,j} \geq 0,\ 1 \leq i \leq w,\ 1 \leq j \leq \ell \tag{3}$$

where $A_{i,j}$ indicates the entry at the ith row and jth column of matrix A, and $A_{i,:}$ indicates the ith row of the matrix. The L_1 constraint encourages sparsity in A; λ_1 is a hyperparameter. Equation 2 constrains D to eliminate solutions where the elements of A are made arbitrarily small by making the norm of D arbitrarily large. Equation 3 ensures that A is non-negative. Together, A and D factor the original corpus statistics matrix X to minimize reconstruction error. One may tune ℓ and λ_1 to vary the sparsity of the final solution.

Murphy et al. (2012) solved this system of constraints using the Online Dictionary Learning algorithm described in Mairal et al. (2010). Though Equations 1-3 represent a non-convex system, when solving for A with D fixed (and vice versa) the loss function is convex. Mairal et al. break the problem into two alternating optimization steps (solving for A and D) and find the system converges to a stationary solution. The solution for A is found with a LARS implementation for lasso regression (Efron et al., 2004); D is found via gradient descent. Though the final solution may not be globally optimal, this method is capable of handling large amounts of data and has been shown to produce useful solutions in practice (Mairal et al., 2010; Murphy et al., 2012).

2.2 Compositional NNSE

We add an additional constraint to the NNSE loss function that allows us to learn a latent representation that respects the notion of semantic composition. As we will see, this change to the loss function has a huge effect on the learned latent space. Just as

Table 1: CNNSE interpretable summarizations for the top 3 dimensions of an adjective, noun and adjective-noun phrase.

military	aid	military aid (observed)
servicemen, commandos, military intelligence	guidance, advice, assistance	servicemen, commandos, military intelligence
guerrilla, paramilitary, anti-terrorist	mentoring, tutoring, internships	guidance, advice, assistance
conglomerate, giants, conglomerates	award, awards, honors	compliments, congratulations, replies

the L_1 regularizer can have a large impact on sparsity, our composition constraint represents a considerable change in composition compatibility.

Consider a phrase p made up of words i and j. In the most general setting, the following composition constraint could be applied to the rows of matrix A corresponding to p, i and j:

$$A_{(p,:)} = f(A_{(i,:)}, A_{(j,:)}) \tag{4}$$

where f is some composition function. The composition function constrains the space of learned latent representations $A \in \mathbb{R}^{w \times \ell}$ to be those solutions that are compatible with the composition function defined by f. Incorporating f into Equation 1 we have:

$$\underset{A,D,\Omega}{\operatorname{argmin}} \sum_{i=1}^{w} \frac{1}{2}\left\|X_{i,:} - A_{i,:} \times D\right\|^2 + \lambda_1 \left\|A_{i,:}\right\|_1 + \frac{\lambda_c}{2} \sum_{\substack{\text{phrase } p, \\ p=(i,j)}} \left(A_{(p,:)} - f(A_{(i,:)}, A_{(j,:)})\right)^2 \tag{5}$$

Where each phrase p is comprised of words (i, j) and Ω represents all parameters of f to be optimized. We have added a squared loss term for composition, and a new regularization parameter λ_c to weight the importance of respecting composition. We call this new formulation Compositional Non-Negative Sparse Embeddings (CNNSE). Some examples of the interpretable representations learned by CNNSE for adjectives, nouns and phrases appear in Table 1.

There are many choices for f: addition, multiplication, dilation, etc. (Mitchell and Lapata, 2010). Here we choose f to be weighted addition because it has has been shown to work well for adjective noun composition (Mitchell and Lapata, 2010; Dinu et al., 2013; Hashimoto et al., 2014), and because it lends itself well to optimization. Weighted addition is:

$$f(A_{(i,:)}, A_{(j,:)}) = \alpha A_{(i,:)} + \beta A_{(j,:)} \tag{6}$$

This choice of f requires that we simultaneously optimize for A, D, α and β. However, α and β are simply constant scaling factors for the vectors in A corresponding to adjectives and nouns. For adjective-noun composition, the optimization of α and β can be absorbed by the optimization of A. For models that include noun-noun composition, if α and β are assumed to be absorbed by the optimization of A, this is equivalent to setting $\alpha = \beta$.

We can further simplify the loss function by constructing a matrix B that imposes the composition by addition constraint. B is constructed so that for each phrase $p = (i, j)$: $B_{(p,p)} = 1$, $B_{(p,i)} = -\alpha$, and $B_{(p,j)} = -\beta$. For our models, we use $\alpha = \beta = 0.5$, which serves to average the single word representations. The matrix B allows us to reformulate the loss function from Eq 5:

$$\underset{A,D}{\operatorname{argmin}} \frac{1}{2}\left\|X - AD\right\|_F^2 + \lambda_1 \left\|A\right\|_1 + \frac{\lambda_c}{2}\left\|BA\right\|_F^2 \tag{7}$$

where F indicates the Frobenius norm. B acts as a selector matrix, subtracting from the latent representation of the phrase the average latent representation of the phrase's constituent words.

We now have a loss function that is the sum of several convex functions of A: squared reconstruction loss for A, L_1 regularization and the composition constraint. This sum of sub-functions is the format required for the alternating direction method of multipliers (ADMM) (Boyd, 2010). ADMM substitutes a dummy variable z for A in the sub-functions:

$$\underset{A,D}{\operatorname{argmin}} \frac{1}{2}\left\|X - AD\right\|_F^2 + \lambda_1 \left\|z_1\right\|_1 + \frac{\lambda_c}{2}\left\|Bz_c\right\|_F^2 \tag{8}$$

and, in addition to constraints in Eq 2 and 3, incorporates constraints $A = z_1$ and $A = z_c$ to ensure dummy variables match A. ADMM uses an aug-

mented Lagrangian to incorporate and relax these new constraints. We optimize for A, z_1 and z_c separately, update the dual variables and repeat until convergence (see Supplementary material for Lagrangian form, solutions and updates). We modified code for ADMM, which is available online[1]. ADMM is used when solving for A in the Online Dictionary Learning algorithm, solving for D remains unchanged from the NNSE implementation (see Algorithms 1 and 2 in Supplementary Material).

We use the weighted addition composition function because it performed well for adjective-noun composition in previous work (Mitchell and Lapata, 2010; Dinu et al., 2013; Hashimoto et al., 2014), maintains the convexity of the loss function, and is easy to optimize. In contrast, an element-wise multiplication, dilation or higher-order matrix composition function will lead to a non-convex optimization problem which cannot be solved using ADMM. Though not explored here, we hypothesize that A could be molded to respect many different composition functions. However, if the chosen composition function does not maintain convexity, finding a suitable solution for A may prove challenging. We also hypothesize that even if the chosen composition function is not the "true" composition function (whatever that may be), the fact that A can change to suit the composition function may compensate for this mismatch. This has the flavor of variational inference for Bayesian methods: an approximation in place of an intractable problem often yields better results with limited data, in less time.

3 Data and Experiments

We use the semantic vectors made available by Fyshe et al. (2013), which were compiled from a 16 billion word subset of ClueWeb09 (Callan and Hoy, 2009). We used the 1000 dependency SVD dimensions, which were shown to perform well for composition tasks. Dependency features are tuples consisting of two POS tagged words and their dependency relationship in a sentence; the feature value is the pointwise positive mutual information (PPMI) for the tuple. The dataset is comprised of 54,454 words and phrases. We randomly split the approximately 14,000 adjective noun phrases into a train (2/3) and test (1/3) set. From the test set we removed 200 randomly selected phrases as a development set for parameter tuning. We did not lexically split the train and test sets, so many words appearing in training phrases also appear in test phrases. For this reason we cannot make specific claims about the generalizability of our methods to unseen words.

[1] http://www.stanford.edu/~boyd/papers/admm/

Table 2: Median rank, mean reciprocal rank (MRR) and percentage of test phrases ranked perfectly (i.e. first in a sorted list of approx. 4,600 test phrases) for four methods of estimating the test phrase vectors. w.add$_{\text{SVD}}$ is weighted addition of SVD vectors, w.add$_{\text{NNSE}}$ is weighted addition of NNSE vectors.

Model	Med. Rank	MRR	Perfect
w.add$_{\text{SVD}}$	99.89	35.26	20%
w.add$_{\text{NNSE}}$	99.80	28.17	16%
Lexfunc	99.65	28.96	20%
CNNSE	**99.91**	**40.65**	**26%**

NNSE has one parameter to tune (λ_1); CNNSE has two: λ_1 and λ_c. In general, these methods are not overly sensitive to parameter tuning, and searching over orders of magnitude will suffice. We found the optimal settings for NNSE were $\lambda_1 = 0.05$, and for CNNSE $\lambda_1 = 0.05, \lambda_c = 0.5$. Too large λ_1 leads to overly sparse solutions, too small reduces interpretability. We set $\ell = 1000$ for both NNSE and CNNSE and altered sparsity by tuning only λ_1.

3.1 Phrase Vector Estimation

To test the ability of each model to estimate phrase semantics we trained models on the training set, and used the learned model and the composition function to estimate vectors of held out phrases. We sort the vectors for the test phrases, X_{test}, by their cosine distance to the predicted phrase vector $\hat{X}_{(p,:)}$.

We report two measures of accuracy. The first is median rank accuracy. Rank accuracy is: $100 \times (1 - \frac{r}{P})$, where r is the position of the correct phrase in the sorted list of test phrases, and $P = |X_{test}|$ (the number of test phrases). The second measure is mean reciprocal rank (MRR), which is often used to evaluate information retrieval tasks (Kantor and Voorhees, 2000). MRR is

$$100 \times (\frac{1}{P} \sum_{i=1}^{P} (\frac{1}{r})). \quad (9)$$

For both rank accuracy and MRR, a perfect score is 100. However, MRR places more emphasis on ranking items close to the top of the list, and less on differences in ranking lower in the list. For example, if the correct phrase is always ranked 2, 50 or 100 out of list of 4600, median rank accuracy would be 99.95, 98.91 or 97.83. In contrast, MRR would be 50, 2 or 1. Note that rank accuracy and reciprocal rank produce identical orderings of methods. That is, whatever method performs best in terms of rank accuracy will also perform best in terms of reciprocal rank. MRR simply allows us to discriminate between very accurate models. As we will see, the rank accuracy of all models is very high ($> 99\%$), approaching the rank accuracy ceiling.

3.1.1 Estimation Methods

We will compare to two other previously studied composition methods: weighted addition (**w.add$_{\text{SVD}}$**), and **lexfunc** (Baroni and Zamparelli, 2010). Weighted addition finds α, β to optimize

$$(X_{(p,:)} - (\alpha X_{(i,:)} + \beta X_{(j,:)}))^2$$

Note that this optimization is performed over the SVD matrix X, rather than on A. To estimate X for a new phrase $p = (i, j)$ we compute

$$\hat{X}_{(p,:)} = \alpha X_{(i,:)} + \beta X_{(j,:)}$$

Lexfunc finds an adjective-specific matrix M_i that solves

$$X_{(p,:)} = M_i X_{(j,:)}$$

for all phrases $p = (i, j)$ for adjective i. We solved each adjective-specific problem with Matlab's partial least squares implementation, which uses the SIMPLS algorithm (Dejong, 1993). To estimate X for a new phrase $p = (i, j)$ we compute

$$\hat{X}_{(p,:)} = M_i X_{(j,:)}$$

We also optimized the weighted addition composition function over NNSE vectors, which we call **w.add$_{\text{NNSE}}$**. After optimizing α and β using the training set, we compose the latent word vectors to estimate the held out phrase:

$$\hat{A}_{(p,:)} = \alpha A_{(i,:)} + \beta A_{(j,:)}$$

For CNNSE, as in the loss function, $\alpha = \beta = 0.5$ so that the average of the word vectors approximates the phrase.

$$\hat{A}_{(p,:)} = 0.5 \times (A_{(i,:)} + A_{(j,:)})$$

Crucially, w.add$_{\text{NNSE}}$ estimates α, β *after* learning the latent space A, whereas CNNSE *simultaneously* learns the latent space A, while taking the composition function into account. Once we have an estimate $\hat{A}_{(p,:)}$ we can use the NNSE and CNNSE solutions for D to estimate the corpus statistics X.

$$\hat{X}_{(p,:)} = \hat{A}_{(p,:)} D$$

Results for the four methods appear in Table 2. Median rank accuracies were all within half a percentage point of each other. However, MRR shows a striking difference in performance. CNNSE has MRR of 40.64, more than 5 points higher than the second highest MRR score belonging to w.add$_{\text{SVD}}$ (35.26). CNNSE ranks the correct phrase in the first position for 26% of phrases, compared to 20% for w.add$_{\text{SVD}}$. Lexfunc ranks the correct phrase first for 20% of the test phrases, w.add$_{\text{NNSE}}$ 16%. So, while all models perform quite well in terms of rank accuracy, when we use the more discriminative MRR, CNNSE is the clear winner. Note that the performance of w.add$_{\text{NNSE}}$ is much lower than CNNSE. Incorporating a composition constraint into the learning algorithm has produced a latent space that surpasses all methods tested for this task.

We were surprised to find that lexfunc performed relatively poorly in our experiments. Dinu et al. (2013) used simple unregularized regression to estimate M. We also replicated that formulation, and found phrase ranking to be worse when compared to the Partial Least Squares method described in Baroni and Zamparelli (2010). In addition, Baroni and Zamparelli use 300 SVD dimensions to estimate M. We found that, for our dataset, using all 1000 dimensions performed slightly better.

We hypothesize that our difference in performance could be due to the difference in input corpus statistics (in particular the thresholding of infrequent words and phrases), or due to the fact that we did not specifically create the training and tests sets to evenly distribute the phrases for each adjective. If an adjective i appears only in phrases in the test set, lexfunc cannot estimate M_i using training data (a hindrance not present for other methods, which

require only that the adjective appear in the training data). To compensate for this possibly unfair train/test split, the results in Table 2 are calculated over only those adjectives which could be estimated using the training set.

Though the results reported here are not as high as previously reported, lexfunc was found to be only slightly better than w.add$_{SVD}$ for adjective noun composition (Dinu et al., 2013). CNNSE outperforms w.add$_{SVD}$ by a large margin, so even if Lexfunc could be tuned to perform at previous levels on this dataset, CNNSE would likely still dominate.

3.1.2 Phrase Estimation Errors

None of the models explored here are perfect. Even the top scoring model, CNNSE, only identifies the correct phrase for 26% of the test phrases. When a model makes a "mistake", it is possible that the top-ranked phrase is a synonym of, or closely related to, the actual phrase. To evaluate mistakes, we chose test phrases for which all 4 models are incorrect and produce a different top ranked phrase (likely these are the most difficult phrases to estimate). We then asked Mechanical Turk (Mturk `http://mturk.com`) users to evaluate the mistakes. We presented the 4 mistakenly top-ranked phrases to Mturk users, who were asked to choose the one phrase most related to the actual test phrase.

We randomly selected 200 such phrases and asked 5 Mturk users to evaluate each, paying $0.01 per answer. We report here the results for questions where a majority (3) of users chose the same answer (82% of questions). For all Mturk experiments described in this paper, a screen shot of the question appears in the Supplementary Material.

Table 3 shows the Mturk evaluation of model mistakes. CNNSE and lexfunc make the most reasonable mistakes, having their top-ranked phrase chosen as the most related phrase 35.4% and 31.7% of the time, respectively. This makes us slightly more comfortable with our phrase estimation results (Table 2); though lexfunc does not reliably predict the correct phrase, it often chooses a close approximation. The mistakes from CNNSE are chosen slightly more often than lexfunc, indicating that CNNSE also has the ability to reliably predict the correct phrase, or a phrase deemed more related than those chosen by other methods.

Table 3: A comparison of mistakes in phrase ranking across 4 composition methods. To evaluate mistakes, we chose phrases for which all 4 models rank a different (incorrect) phrase first. Mturk users were asked to identify the phrase that was semantically closest to the target phrase.

Model	Predicted phrase deemed closest match to actual phrase
w.add$_{SVD}$	21.3%
w.add$_{NNSE}$	11.6%
Lexfunc	31.7%
CNNSE	**35.4%**

3.2 Interpretability

Though our improvement in MRR for phrase vector estimation is compelling, we seek to explore the meaning encoded in the word space features. We turn now to the *interpretation* of phrasal semantics and semantic composition.

3.2.1 Interpretability of Latent Dimensions

Due to the sparsity and non-negativity constraints, NNSE produces dimensions with very coherent semantic groupings (Murphy et al., 2012). Murphy et al. used an intruder task to quantify the interpretability of semantic dimensions. The intruder task presents a human user with a list of words, and they are to choose the one word that does not belong in the list (Chang et al., 2009). For example, from the list (red, green, desk, pink, purple, blue), it is clear to see that the word "desk" does not belong in the list of colors.

To create questions for the intruder task, we selected the top 5 scoring words in a particular dimension, as well as a low scoring word from that same dimension such that the low scoring word is also in the top 10th percentile of another dimension. Like the word "desk" in the example above, this low scoring word is called the *intruder*, and the human subject's task is to select the intruder from a shuffled list of 6 words. Five Mturk users answered each question, each paid $0.01 per answer. If Mturk users identify a high percentage of intruders, this indicates that the latent representation groups words in a human-interpretable way. We chose 100 questions for each of the NNSE, CNNSE and SVD representations. Because the output of lexfunc is the SVD

Table 4: Quantifying the interpretability of learned semantic representations via the intruder task. Intruders detected: % of questions for which the majority response was the intruder. Mturk agreement: the % of questions for which a majority of users chose the same response.

Method	Intruders Detected	Mturk Agreement
SVD	17.6%	74%
NNSE	86.2%	94%
CNNSE	88.9%	90%

representation X, SVD interpretability is a proxy for lexfunc interpretability.

Results for the intruder task appear in Table 4. Consistent with previous studies, NNSE provides a much more interpretable latent representation than SVD. We find that the additional composition constraint used in CNNSE has maintained the interpretability of the learned latent space. Because intruders detected is higher for CNNSE, but agreement amongst Mturk users is higher for NNSE, we consider the interpretability results for the two methods to be equivalent. Note that SVD interpretability is close to chance ($1/6 = 16.7\%$).

3.2.2 Coherence of Phrase Representations

The dimensions of NNSE and CNNSE are comparably interpretable. But, has the composition constraint in CNNSE resulted in better phrasal representations? To test this, we randomly selected 200 phrases, and then identified the top scoring dimension for each phrase in both the NNSE and CNNSE models. We presented Mturk users with the interpretable summarizations for these top scoring dimensions. Users were asked to select the list of words (interpretable summarization) most closely related to the target phrase. Mturk users could also select that neither list was related, or that the lists were equally related to the target phrase. We paid \$0.01 per answer and had 5 users answer each question. In Table 5 we report results for phrases where the majority of users selected the same answer (78% questions). CNNSE phrasal representations are found to be much more consistent, receiving a positive evaluation almost twice as often as NNSE.

Together, these results show that CNNSE representations maintain the interpretability of NNSE dimensions, while improving the coherence of phrase representations.

Table 5: Comparing the coherence of phrase representations from CNNSE and NNSE. Mturk users were shown the interpretable summarization for the top scoring dimension of target phrases. Representations from CNNSE and NNSE were shown side by side and users were asked to choose the list (summarization) most related to the phrase, or that the lists were equally good or bad.

Model	Model representation deemed most consistent with phrase
CNNSE	**54.5%**
NNSE	29.5%
Both	4.5%
Neither	11.5%

3.3 Evaluation on Behavioral Data

We now compare the performance of various composition methods on an adjective-noun phrase similarity dataset (Mitchell and Lapata, 2010). This dataset is comprised of 108 adjective-noun phrase pairs split into high, medium and low similarity groups. Similarity scores from 18 human subjects are averaged to create one similarity score per phrase pair. We then compute the cosine similarity between the composed phrasal representations of each phrase pair under each compositional model. As in Mitchell and Lapata (2010), we report the correlation of the cosine similarity measures to the behavioral scores. We withheld 12 of the 108 questions for parameter tuning, four randomly selected from each of the high, medium and low similarity groups.

Table 6 shows the correlation of each model's similarity scores to behavioral similarity scores. Again, Lexfunc performs poorly. This is probably attributable to the fact that there are, on average, only 39 phrases available for training each adjective in the dataset, whereas the original Lexfunc study had at least 50 per adjective (Baroni and Zamparelli, 2010). CNNSE is the top performer, followed closely by weighted addition. Interestingly, weighted NNSE correlation is lower than CNNSE by nearly 0.15, which shows the value of allowing the learned latent space to conform to the desired composition function.

3.3.1 Interpretability and Phrase Similarity

CNNSE has the additional advantage of interpretability. To illustrate, we created a web page to explore the dataset under the CNNSE model. The page `http://www.cs.cmu.edu/~fmri/papers/naacl2015/cnnse_mitchell_lapata_all.html` displays phrase pairs sorted by average similarity score. For each phrase in the pair we show a summary of the CNNSE composed phrase meaning. The scores of the 10 top dimensions are displayed in descending order. Each dimension is described by its interpretable summarization. As one scrolls down the page, the similarity scores increase, and the number of dimensions shared between the phrase pairs (highlighted in red) increases. Some phrase pairs with high similarity scores share no top scoring dimensions. Because we can interpret the dimensions, we can begin to understand how the CNNSE model is failing, and how it might be improved.

For example, the phrase pair judged most similar by the human subjects, but that shares none of the top 10 dimensions in common, is "large number" and "great majority" (behavioral similarity score 5.61/7). Upon exploration of CNNSE phrasal representations, we see that the representation for "great majority" suffers from the multiple word senses of majority. Majority is often used in political settings to describe the party or group with larger membership. We see that the top scoring dimension for "great majority" has top scoring words "candidacy, candidate, caucus", a politically-themed dimension. Though the CNNSE representation is not incorrect for the word, the common theme between the two test phrases is not political.

The second highest scoring dimension for "large number" is "First name, address, complete address". Here we see another case of the collision of multiple word senses, as this dimension is related to identifying numbers, rather than the quantity-related sense of number. While it is satisfying that the word senses for majority and number have been separated out into different dimensions for each word, it is clear that both the composition and similarity functions used for this task are not gracefully handling multiple word senses. To address this issue, we could partition the dimensions of A into sense-related groups and use the maximally correlated groups to score phrase pairs. CNNSE interpretability allows us to perform these analyses, and will also allow us to iterate and improve future compositional models.

Table 6: Correlation of phrase similarity judgements (Mitchell and Lapata, 2010) to pairwise distances in several adjective-noun composition models.

Model	Correlation to behavioral data
w.add$_{\text{SVD}}$	0.5377
w.add$_{\text{NNSE}}$	0.4469
Lexfunc	0.1347
CNNSE	**0.5923**

4 Conclusion

We explored a new method to create an interpretable VSMs that respects the notion of semantic composition. We found that our technique for incorporating phrasal relationship constraints produced a VSM that is more consistent with observed phrasal representations and with behavioral data.

We found that, compared to NNSE, human evaluators judged CNNSE phrasal representations to be a better match to phrase meaning. We leveraged this improved interpretability to explore composition in the context of a previously published compositional task. We note that the collision of word senses often hinders performance on the behavioral data from Mitchell and Lapata (2010).

More generally, we have shown that incorporating constraints to represent the task of interest can improve a model's performance on that task. Additionally, incorporating such constraints into an *interpretable* model allows for a deeper exploration of performance in the context of evaluation tasks.

Acknowledgments

This work was supported in part by a gift from Google, NIH award 5R01HD075328, IARPA award FA865013C7360, DARPA award FA8750-13-2-0005, and by a fellowship to Alona Fyshe from the Multimodal Neuroimaging Training Program (NIH awards T90DA022761 and R90DA023420).

References

Marco Baroni and Roberto Zamparelli. Nouns are vectors, adjectives are matrices: Representing

adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193. Association for Computational Linguistics, 2010.

Stephen Boyd. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010. ISSN 1935-8237. doi: 10.1561/2200000016.

Jamie Callan and Mark Hoy. The ClueWeb09 Dataset, 2009. URL `http://boston.lti.cs.cmu.edu/Data/clueweb09/`.

Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M Blei. Reading Tea Leaves : How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems*, pages 1–9, 2009.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

S Dejong. SIMPLS - An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18(3):251–263, 1993. ISSN 01697439. doi: 10.1016/0169-7439(93)85002-x.

Georgiana Dinu, Nghia The Pham, and Marco Baroni. General estimation and evaluation of compositional distributional semantic models. In *Workshop on Continuous Vector Space Models and their Compositionality*, Sofia, Bulgaria, 2013.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

Alona Fyshe, Partha Talukdar, Brian Murphy, and Tom Mitchell. Documents and Dependencies : an Exploration of Vector Space Models for Semantic Composition. In *Computational Natural Language Learning*, Sofia, Bulgaria, 2013.

Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. Jointly learning word representations and composition functions using predicate-argument structures. *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 1544–1555, 2014.

Paul B. Kantor and Ellen M. Voorhees. The TREC-5 Confusion Track: Comparing Retrieval Methods for Scanned Text. *Information Retrieval*, 2:165–176, 2000. ISSN 1386-4564, 1573-7659. doi: 10.1023/A:1009902609570.

Omer Levy and Yoav Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems*, pages 1–9, 2014.

Julien Mairal, Francis Bach, J Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of Neural Information Processing Systems*, pages 1–9, 2013.

Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–429, November 2010. ISSN 1551-6709. doi: 10.1111/j.1551-6709.2010.01106.x.

Brian Murphy, Partha Talukdar, and Tom Mitchell. Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding. In *Proceedings of Conference on Computational Linguistics (COLING)*, 2012.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe : Global Vectors for Word Representation. In *Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014.

Magnus Sahlgren. *The Word-Space Model Using distributional analysis to represent syntagmatic and paradigmatic relations between words*. Doctor of philosophy, Stockholm University, 2006.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic Compositionality through Recursive Matrix-Vector Spaces. In *Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012.

Peter D Turney. Domain and Function : A Dual-Space Model of Semantic Relations and Compositions. *Journal of Artificial Intelligence Research*, 44:533–585, 2012.