Documents and Dependencies: an Exploration of Vector Space Models for Semantic Composition

Alona Fyshe, Partha Talukdar, Brian Murphy and Tom Mitchell
Machine Learning Department &
Center for the Neural Basis of Cognition
Carnegie Mellon University, Pittsburgh
{afyshe|ppt|bmurphy|tom.mitchell}@cs.cmu.edu

Abstract

In most previous research on distributional semantics, Vector Space Models (VSMs) of words are built either from *topical* information (e.g., documents in which a word is present), or from syntactic/semantic *types* of words (e.g., dependency parse links of a word in sentences), but not both. In this paper, we explore the utility of combining these two representations to build VSM for the task of semantic composition of adjective-noun phrases. Through extensive experiments on benchmark datasets, we find that even though a type-based VSM is effective for semantic composition, it is often outperformed by a VSM built using a combination of topic- and type-based statistics. We also introduce a new evaluation task wherein we predict the composed vector representation of a phrase from the brain activity of a human subject reading that phrase. We exploit a large syntactically parsed corpus of 16 billion tokens to build our VSMs, with vectors for both phrases and words, and make them publicly available.

1 Introduction

Vector space models (VSMs) of word semantics use large collections of text to represent word meanings. Each word vector is composed of features, where features can be derived from global corpus co-occurrence patterns (e.g. how often a word appears in each document), or local corpus co-occurrence patterns patterns (e.g. how often two words appear together in the same sentence, or are linked together in dependency parsed sentences). These two feature types represent different aspects of word meaning (Murphy et al., 2012c), and can be compared with the paradigmatic/syntagmatic distinction (Sahlgren, 2006). Global patterns give a more *topic-based* meaning (e.g. *judge* might appear in documents also containing *court* and *verdict*). Certain local patterns give a more *type-based* meaning (e.g. the noun *judge* might be modified by the adjective *harsh*, or be the subject of *decide*, as would related and substitutable words such as *referee* or *conductor*). Global patterns have been used in Latent Semantic Analysis (Landauer and Dumais, 1997) and LDA Topic models (Blei et al., 2003). Local patterns based on word co-occurrence in a fixed width window were used in Hyperspace Analogue to Language (Lund and Burgess, 1996). Subsequent models added increasing linguistic sophistication, up to full syntactic and dependency parses (Lin, 1998; Padó and Lapata, 2007; Baroni and Lenci, 2010).

In this paper we systematically explore the utility of a global, topic-based VSM built from what we call *Document* features, and a local, type-based VSM built from *Dependency* features. Our Document VSM represents each word w by a vector where each feature is a specific document, and the feature value is the number of mentions of word w in that document. Our Dependency VSM represents word w with a vector where each feature is a dependency parse link (e.g., the word w is the subject of the verb "eat"), and the feature value is the number of instances of this dependency feature for word w across a large text corpus. We also consider a third *Combined* VSM in which the word vector is the concatenation of its Document and Dependency features. All three models subsequently normalize frequencies using positive pointwise mutual-information (PPMI), and are dimensionality reduced using singular value decomposition (SVD). This is the first systematic study of the utility of Document and Dependency features for semantic composition. We construct all three VSMs (Dependencies, Documents, Combined) using the same text corpus and preprocessing pipeline, and make the resulting VSMs available for download (`http://www.cs.cmu.edu/~afyshe/papers/conll2013/`). To our knowledge, this is the first freely available VSM that includes entries for both words and adjective-noun phrases, and it is built from a much larger corpus than previously shared resources (16 billion words, 50 million documents). Our main contributions include:

- We systematically study complementarity of topical (Document) and type (Dependency) features in Vector Space Model (VSM) for semantic composition of adjective-noun phrases. To the best of our knowledge, this is one of the first studies of this kind.
- Through extensive experiments on standard benchmark datasets, we find that a VSM built from a combination of topical and type features is more effective for semantic composition, compared to a VSM built from Document and Dependency features alone.
- We introduce a novel task: to predict the vector representation of a composed phrase from the brain activity of human subjects reading that phrase.
- We explore two composition methods, addition and dilation, and find that while addition performs well on corpus-only tasks, dilation performs best on the brain activity task.
- We build our VSMs, for both phrases and words, from a large syntactically parsed text corpus of 16 billion tokens. We also make the resulting VSM publicly available.

2 Related Work

Mitchell and Lapata (2010) explored several methods of combining adjective and noun vectors to estimate phrase vectors, and compared the similarity judgements of humans to the similarity of their predicted phrase vectors. They found that for adjective-noun phrases, type-based models outperformed Latent Dirichlet Allocation (LDA) topic models. For the type-based models, multiplication performed the best, followed by weighted addition and a dilation model (for details on composition functions see Section 4.2). However, Mitchell and Lapata did not combine the topic- and type-based models, an idea we explore in detail in this paper.

Baroni and Zamparelli (2010) extended the typical vector representation of words. Their model used matrices to represent adjectives, while nouns were represented with column vectors. The vectors for nouns and adjective-noun phrases were derived from local word co-occurrence statistics. The matrix to represent the adjective was estimated with partial least squares regression where the product of the learned adjective matrix and the observed noun vector should equal the observed adjective-noun vector. Socher et al. (2012) also extended word representations beyond simple vectors. Their model assigns each word a vector and a matrix, which are composed via an non-linear function (e.g. tanh) to create phrase representations consisting of another vector/matrix pair. This process can proceed recursively, following a parse tree to produce a composite sentence meaning. Other general semantic composition frameworks have been suggested, e.g. (Sadrzadeh and Grefenstette, 2011) who focus on the operational nature of composition, rather than the representations that are supplied to the framework. Here we focus on creating word representations that are useful for semantic composition.

Turney (2012) published an exploration of the impact of domain- and function-specific vector space models, analogous to the topic and type meanings encoded by our Document and Dependency models respectively. In Turney's work, domain-specific information was represented by noun token co-occurrence statistics within a local window, and functional roles were represented by generalized token/part-of-speech co-occurrence patterns with verbs - both of which are relatively local and shallow when compared with this work. Similar local context-based features were used to cluster phrases in (Lin and Wu, 2009). Though the models discussed here are not entirely comparable to it, a recent comparison suggested that broader, deeper features such as ours may result in representations that are superior for tasks involving neural activation data (Murphy et al., 2012b).

In contrast to the composite model in (Griffiths et al., 2005), in this paper we explore the complementarity of semantics captured by topical information and syntactic/semantic types. We focus on learning VSMs (involving both words and phrases) for semantic composition, and use more expressive dependency-based features in our type-based VSM. A comparison of vector-space representations was recently published (Blacoe and Lapata, 2012), in which the authors compared several methods of combining single words vectors to create phrase vectors. They found that the best performance for adjective-noun composition used point-wise multiplication and a model based on type-based word co-occurrence patterns.

3 Creating a Vector-Space

To create the Dependency vectors, a 16 billion word subset of ClueWeb09 (Callan and Hoy, 2009) was dependency parsed using the Malt parser (Hall et al., 2007). Dependency statistics were then collected for a predetermined list of target words and adjective-noun phrases, and for arbitrary adjective-noun phrases observed in the corpus. The list was composed of the 40 thousand most frequent single tokens in the American National Corpus (Ide and Suderman, 2006), and a small number of words and phrases used as stimuli in our brain imaging experiments. Additionally, we included any phrase found in the corpus whose maximal token span matched the PoS pattern *J*+*N*+, where *J* and *N* denote adjective and noun PoS tags respectively. For each *unit* (i.e., word or phrase) in this augmented list, counts of all unit-external dependencies incident on the head word were aggregated across the corpus, while unit-internal dependencies were ignored. Each token was appended with its PoS tag, and the dependency edge label was also included. This resulted in the extraction of 498 million dependency tuples. For example, the dependency tuple *(a/DT, NMOD, 27-inch/JJ television/NN,14)*, indicates that *a/DT* was found as a child of *27-inch/JJ television/NN* with a frequency of 14 in the corpus.

To create Document vectors, word-document co-occurrence counts were taken from the same subset of Clueweb, which covered 50 million documents. We applied feature-selection for computational efficiency reasons, ranking documents by the number of target word/phrase types they contained and choosing the top 10 million.

A series of three additional filtering steps selected target words/phrases, and Document/Dependency features for which there was adequate data.[1] First, a co-occurrence frequency cut-off was used to reduce the dimensionality of the matrices, and to discard noisy estimates. A cutoff of 20 was applied to the dependency counts, and of 2 to document counts. Positive pointwise-mutual-information (PPMI) was used as an association measure to normalize the observed co-occurrence frequency for the varying frequency of the target word and its features, and to discard negative associations. Second, the target list was filtered to the 57 thousand words and phrases which had at least 20 non-“stop word” Dependency co-occurrence types, where a “stop word” was one of the 100 most frequent Dependency features observed (so named because the dependencies were largely incident on function words). Third, features observed for no more than one target were removed, as were empty target entries. The result was a Document co-occurrence matrix of 55 thousand targets by 5.2 million features (total 172 million non-zero entries), and a Dependency matrix of 57 thousand targets by 1.25 million features (total 35 million non-zero entries).

A singular value decomposition (SVD) matrix factorization was computed separately on the Dependency and Document statistics matrices, with 1000 latent dimensions retained. For this step we used Python/Scipy implementation of the Implicitly Restarted Arnoldi method (Lehoucq et al., 1998; Jones et al., 2001). This method is compatible with PPMI normalization, since a zero value represents both negative target-feature associations, and those that were not observed or fell below the frequency cut-off. To combine Document and Dependency information, we concatenate vectors.

4 Experiments

To evaluate how Document and Dependency dimensions can interact and compliment each other,

[1] In earlier experiments with more than 500 thousand phrasal entries, we found that the majority of targets were dominated by non-distinctive stop word co-occurrences, resulting in semantically vacuous representations.

Table 1: The nearest neighbors of three queries under three VSMs: all 2000 dimensions (Deps & Docs); 1000 Document dimensions (Docs); 1000 Dependency dimensions (Deps).

Query	Deps & Docs	Docs	Deps
beautiful/JJ	wonderful/JJ lovely/JJ excellent/JJ	wonderful/JJ fantastic/JJ unspoiled/JJ	lovely/JJ gorgeous/JJ wonderful/JJ
dog/NN	cat/NN dogs/NNS pet/NN	dogs/NNS vet/NN leash/NN	cat/NN the/DT_dog/NN dogs/NNS
bad/JJ_publicity/NN	negative/JJ_publicity/NN bad/JJ_press/NN unpleasantness/NN	fast/JJ_cash/NN_loan/NN small/JJ_business/NN_loan/NN important/JJ_cities/NNS	negative/JJ_publicity/NN bad/JJ_press/NN unpleasantness/NN

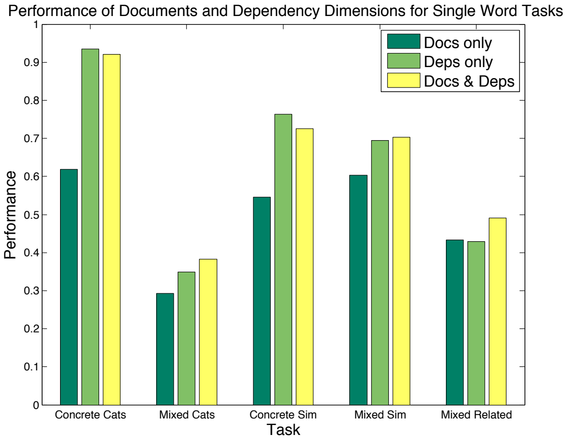


Figure 1: Performance of VSMs for single word behavioral tasks as we vary Document and Dependency inclusion.

we can perform a qualitative comparison between the nearest neighbors (NNs) of words and phrases in the three VSMs – Dependency, Document, and Combined (Dependency & Document). Results appear in Table 1. Note that single words and phrases can be neighbors of each other, demonstrating that our VSMs can generalize across syntactic types. In the Document VSM, we get more topically related words as NNs (e.g., *vet* and *leash* for *dog*); and in the Dependency VSM, we see words that might substitute for one another in a sentence (e.g., *gorgeous* for *beautiful*). The two feature sets can work together to up-weight the most suitable NNs (as in *beautiful*), or help to drown out noise (as in the NNs for *bad publicity* in the Document VSM).

4.1 Judgements of Word Similarity

As an initial test of the informativeness of Document and Dependency features, we evaluate the representation of single words. Behavioral judgement benchmarks have been widely used to evaluate vector space representations (Lund and Burgess, 1996; Rapp, 2003; Sahlgren, 2006). Here we used five such tests. Two tests are categorization tests, where we evaluate how well an automatic clustering of our word vectors correspond to pre-defined word categories. The first "Concrete Categories" test-set consists of 82 nouns, each assigned to one of 10 concrete classes (Battig and Montague, 1969). The second "Mixed Categories" test-set contains 402 nouns in a range of 21 concrete and abstract classes from WordNet (Almuhareb and Poesio, 2004; Miller et al., 1990). Both categorization tests were performed with the Cluto clustering package (Karypis, 2003) using cosine distances. Success was measured as percentage purity over clusters based on their plurality class, with chance performance at 10% and 5% respectively for the "Concrete Categories" and "Mixed Categories" tests.

The remaining three tests use group judgements of similarity: the "Concrete Similarity" set of 65 concrete word pairs (Rubenstein and Goodenough, 1965); and two variations on the WordSim353 test-set (Finkelstein et al., 2002), partitioned into subsets corresponding to strict attributional similarity ("Mixed Similarity", 203 noun pairs), and broader topical "relatedness" ("Mixed Relatedness", 252 noun pairs) (Agirre et al., 2009). Performance on these benchmarks is Spearman correlation between the aggregate human judgements and pairwise cosine distances of word vectors in a VSM.

The results in Figure 1 show that the Dependency VSM substantially outperforms the Document VSM when predicting human judgements of strict attributional (categorial) similarity ("Similarity" as opposed to "Relatedness") for concrete nouns. Conversely the Document VSM is compet-

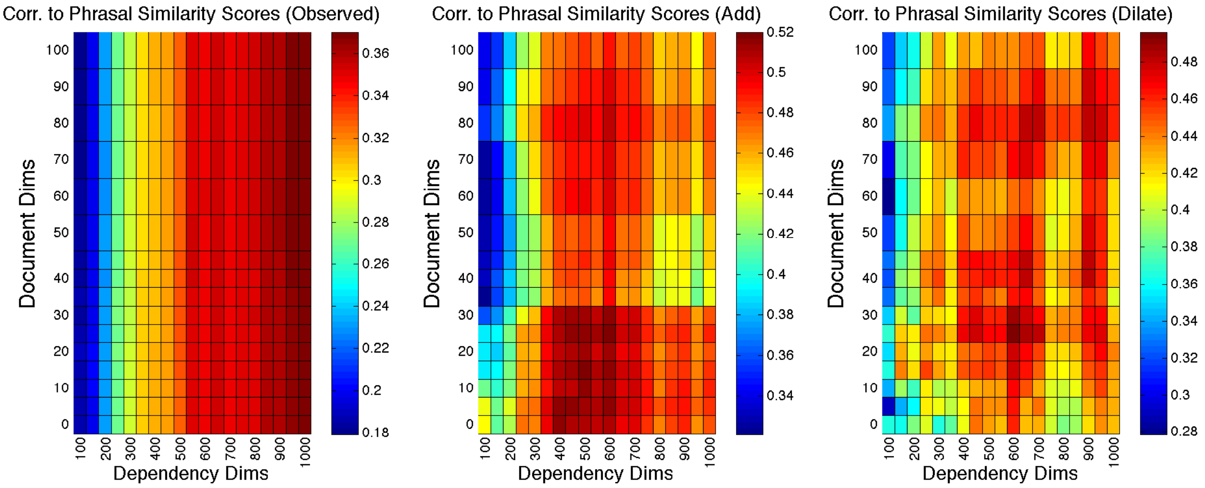


Figure 2: The performance of three phrase representations for predicting the behavioral phrasal similarity scores from Mitchell and Lapata (2010). The highest correlation is 0.5033 and uses 25 Document dimensions, 600 Dependency dimensions and the addition composition function.

itive for less concrete word types, and for judgements of broader topical relatedness.

4.2 Judgements of Phrase Similarity

We also evaluated our system on behavioral data of phrase similarity judgements gathered from 18 human informants. The adjective-noun phrase pairs are divided into 3 groups: high, medium and low similarity (Mitchell and Lapata, 2010). For each pair of phrases, informants rated phrase similarity on a Likert scale of 1-7. There are 36 phrase pairs in each of the three groups for a total of 108 phrase pairs. Not all of the phrases occurred frequently enough in our corpus to pass our thresholds, and so were omitted from our analysis. In several cases we also used pluralizations of the test phrases (e.g.“dark eyes”) where the singular form was not found in our VSM. After these changes we were left with 28, 24 and 28 in the high, medium and low groups respectively. In total we have 80 *observed* vectors for the 108 phrase pairs. These adjective-noun phrases were included in the list of targets, so their statistics were gathered in the same way as for single words. This does not impact results for composed vectors, as all of the single words in the phrases do appear in our VSMs. A full list of the phrase pairs can be found in Mitchell and Lapata (2010).

To evaluate, we used three different representations of phrases. For phrase pairs that passed our thresholds, we can test the similarity of observed representations by comparing the VSM representation of the phrase (no composition function). For all 108 phrase pairs we can test the composed phrase representations, derived by applying addition and dilation operations to word vectors. Multiplication is not used as SVD representations include negative values, and so the product of two negative values would be positive.

Addition is the element-wise sum of two semantic feature vectors $s_i^{add} = s_i^{adj} + s_i^{noun}$, where s_i^{noun}, s_i^{adj}, and s_i^{add} are the i^{th} element of the noun, adjective, and predicted phrase vectors, respectively. Dilation of two semantic feature vectors s^{adj} and s^{noun} is calculated by first decomposing the noun into a component parallel to the adjective (x) and a component perpendicular to the adjective (y) so that $s^{noun} = x + y$. Dilation then enhances the adjective component by multiplying it by a scalar (γ): $s^{dilate} = \gamma x + y$. This can be viewed as taking the representation of the noun, and up-weighting the elements it shares with the adjective, which is coherent with the notion of co-composition (Pustejovsky, 1995). Previous work (Mitchell and Lapata, 2010) tuned the γ parameter ($\gamma = 16.7$). We use that value here, though further optimization might increase performance.

For our evaluation we calculated the cosine distance between pairs of phrases in the three different representation spaces: observed, addition and dilation. Results for a range of dimensionality settings appear in Figure 2. In the observed space, we maximized performance when we included all 1000 of the Document and 350 Dependency dimensions. For consistency the y axis in Figure 2 extends only to 100 Document dimensions: changes beyond 100 dimensions for observed vectors were minimal. By design, SVD will tend to use lower dimensions to represent the strongest signals in the input statistics, which typically originate in the types of targets that are most frequent – in this case single words. We have observed that less frequent and noisier counts, as might be found for many phrases, are displaced to the higher dimensions. Consistent with this observation, maximum performance occurs using a high number of dimensions (correlation of 0.37 to human judgements of phrase similarity).

Interestingly, using the single word vectors to predict the phrase vectors via the addition function gives the best correlation of any of the representations, outperforming even the observed phrase representations. When using 25 Document dimensions and 600 Dependency dimensions the correlation is 0.52, compared to the best performance of 0.51 using Dependency dimensions only. We speculate that the advantage of composed vectors over observed vectors is due to sparseness and resulting noise/variance in the observed phrase vectors, as phrases are necessarily less frequent than their constituent words.

The dilation composition function performs slightly worse than addition, but shows best performance at the same point as addition. Here, the highest correlation (0.46) is substantially lower than that attained by addition, and uses 25 dimensions of the Document, and 600 dimensions of the Dependency VSM.

To summarize, without documents, {observed, addition and dilation} phrase vectors have maximal correlations {0.37, 0.51 and 0.46}. With documents, {observed, addition and dilation} phrase vectors have maximal correlations {0.37, 0.52 and 0.50}. Our results using the addition function (0.52) outperform the results in two previous studies (Mitchell and Lapata, 2010; Blacoe and Lapata, 2012): (0.46 and 0.48 respectively). This is evidence that a VSM built from a larger corpus, and with both Document and Dependency information can yield superior results.

4.3 Composed vs Observed Phrase Vectors

Next we tested how well our representations and semantic composition functions could predict the *observed* vector statistics for phrases from the vectors of their component words. Again, we explored addition and dilation composition functions. For testing we have $13,575$ vectors for which both the adjective and noun passed our thresholds. We predicted a composed phrase vector using the statistics of the single words and one of the two composition functions (addition or dilation). We then sorted the list of *observed* phrase vectors by their distance to the *composed* phrase vector and recorded the position of the corresponding observed vector in the list. From this we calculated percentile rank, the percent of phrases that are further from the predicted vector than the observed vector. Percentile rank is: $100 \times (1 - \mu_{rank}/N)$ where μ_{rank} is the average position of the correct observed vector in the sorted list and $N = 13,575$ is the size of the list.

Figure 3 shows the changes in percentile rank in response to varying dimensions of Documents and Dependencies for the addition function. Dilation results are not shown, but the pattern of performance is very similar. In general, when one includes more Document dimensions, the percentile rank increases. For both the dilation and addition composition functions the peak performance is with 750 Dependency dimensions and 1000 Document dimensions. Dilation's peak performance is 97.87; addition peaks at 98.03 percentile rank. As in Section 4.2, we see that the accurate representation of phrases requires higher SVD dimensions.

To evaluate when composition fails, we examined the cases where the percentile rank was $< 25\%$. Amongst these words we found an over-representation of operational adjectives like "better" and "more". As observed previously, it is possible that such adjectives could be better represented with a matrix or function (Socher et al., 2012; Baroni and Zamparelli, 2010). Composition may also be failing when the adjective-noun phrase is non-compositional (e.g. lazy susan); filtering such phrases could improve performance.

4.4 Brain Activity Data

Here we explore the relationship between the neural activity observed when a person reads a phrase,

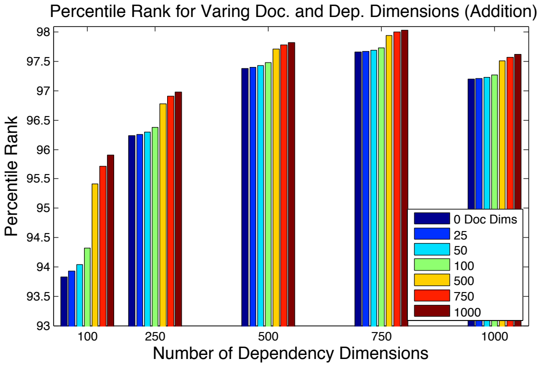
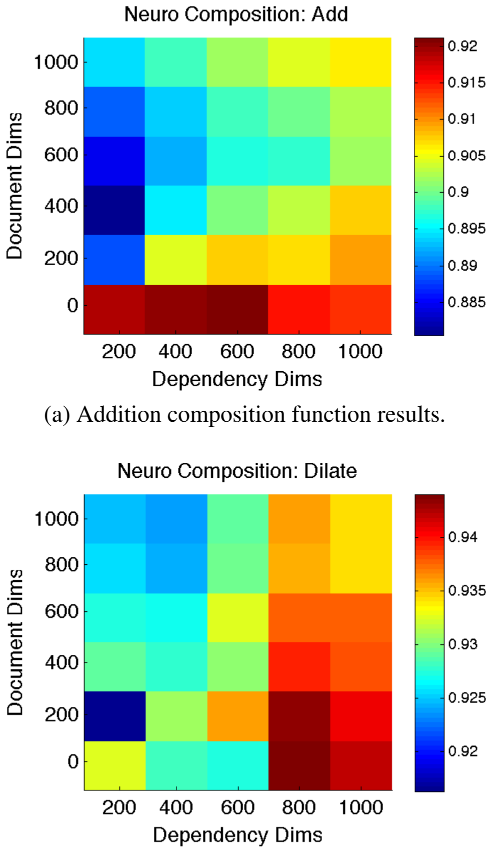


Figure 3: The percentile rank of observed phrase vectors compared to vectors created using the addition composition function.

and our predicted composed VSM for that phrase. We collected brain activity data using Magnetoencephalography (MEG). MEG is a brain imaging method with much higher temporal resolution (1 ms) than fMRI (∼2 sec). Since words are naturally read at a rate of about 2 per second, MEG is a better candidate for capturing the fast dynamics of semantic composition in the brain. Some previous work has explored adjective-noun composition in the brain (Chang et al., 2009), but used fMRI and corpus statistics based only on co-occurrence with 5 hand-selected verbs.

Our MEG data was collected while 9 participants viewed 38 phrases, each repeated 20 times (randomly interleaved). The stimulus nouns were chosen because previous research had shown them to be decodable from MEG recordings, and the adjectives were selected to modulate their most decodable semantic properties (e.g. edibility, manipulability) (Sudre et al., 2012). The 8 adjectives selected are (“big”, “small”, “ferocious”, “gentle”, “light”, “heavy”, “rotten”, “tasty”), and the 6 nouns are (“dog”, “bear”, “tomato”, “carrot”, “hammer”, “shovel”). The words “big” and “small” are paired with every noun, “ferocious” and “gentle” with animals, “light” and “heavy” with tools and “rotten” and “tasty” with foods. We also included the words “the” and the word “thing” as semantically neutral fillers, to present each of the words in a condition without semantic modulation. In total there are 38 phrases (e.g. “rotten carrot”, “big hammer”).

Figure 4: Results for predicting composed phrase vectors (addition [4a] and dilation [4b]) from MEG recordings. Results shown are the average over 9 subjects viewing 38 adjective-noun phrases. This is the one task on which dilation outperforms addition.



(a) Addition composition function results.

(b) Dilation composition function results.

In the MEG experiment, the adjective and paired noun were each shown for 500ms, with a 300ms interval between them, and there were 3 seconds in total time between the onset of subsequent phrases. Data was preprocessed to maximize the signal/noise ratio as is common practice – see Gross et al., (2012). The 20 repeated trials for each phrase were averaged together to create one average brain image per phrase.

To determine if the recorded MEG data can be used to predict our composed vector space representations, we devised the following classification framework.[2] The training data is comprised of the averaged MEG signal for each of the 38 phrases for one subject, and the labels are the 38 phrases. We use our VSMs and composition functions to form a mapping of the 38 phrases to com-

[2] Predicting brain activity from VSM representations is also possible, but provides additional challenges, as parts of the observed brain activity are not driven by semantics.

posed semantic feature vectors $w \rightarrow \{s_1 \ldots s_m\}$. The mapping allows us to use Zero Shot Learning (Palatucci et al., 2009) to predict novel phrases (not seen during training) from a MEG recording. This is a particularly attractive characteristic for the task of predicting words, as there are many words and many more phrases in the English language, and one cannot hope to collect MEG recordings for all of them.

Formally, let us define the semantic representation of a phrase w as semantic feature vector $\vec{s}_w = \{s_1 ... s_m\}$, where the semantic space has dimension m that varies depending on the number of Document and/or Dependency dimensions we include. We utilize the mapping $w \rightarrow \{s_1 \ldots s_m\}$ to train m independent functions $f_1(X) \rightarrow s'_1, \ldots, f_m(X) \rightarrow s'_m$ where s' represents the value of a predicted composed semantic feature. We combine the output of $f_1 \ldots f_m$ to create the final predicted semantic vector $\vec{s'} = \{s'_1 \ldots s'_m\}$. We use cosine distance to quantify the distance between true and predicted semantic vectors.

To measure performance we use the **2 vs. 2 test**. For each test we withhold two phrases and train regressors on the remaining 36. We use the regressors f and MEG data from the two held out phrases to create two predicted semantic vectors. We then choose the assignment of predicted semantic vectors ($\vec{s'_i}$ and $\vec{s'_j}$) to true semantic vectors ($\vec{s_i}$ and $\vec{s_j}$) that minimizes the sum of cosine distances. If we choose the correct assignment ($\vec{s'_i} \mapsto \vec{s_i}$ and $\vec{s'_j} \mapsto \vec{s_j}$) we mark the test as correct. **2 vs. 2 accuracy** is the number of 2 vs. 2 tests with correct assignments divided by the total number of tests. There are (38 choose 2) $=$ 703 distinct 2 vs. 2 tests, and we evaluate on the subset for which neither the adjective nor noun are shared (540 pairs). Chance performance is 0.50.

For each f we trained a regressor with L_2 penalty. We tune the regularization parameter with leave-one-out-cross-validation on training data. We train regressors using the first 800 ms of MEG signal after the noun stimulus appears, when we assume semantic composition is taking place.

Results appear in Figure 4. The best performance (2 vs. 2 accuracy of 0.9440) is achieved with dilation, 800 dimensions of Dependencies and zero Document dimensions. When we use the addition composition function, optimal performance is 0.9212, at 600 Dependency and zero Document dimensions. Note, however, that the parameter search here was much coarser that in Sections 4.2 and 4.3, due to the computation required. We used a finer grid around the peaks in performance for addition and dilation and found minimal improvement ($\pm 0.5\%$) with the addition of a small number of Document dimensions.

It is intriguing that this neurosemantic task is the only task for which dilation outperforms addition. All other composition tasks explored in this study were concerned with matching composed word vectors to observed or composed word vectors, whereas here we are interested in matching composed word vectors to observed *brain activity*. Perhaps the brain works in a manner more akin to the emphasis of elements as modeled by dilation, rather than a summing of features. Further work is required to fully understand this phenomenon, but this is surely a thought-provoking result.[3]

5 Conclusion

We have performed a systematic study of complementarity of topical (Document) and type (Dependency) features in Vector Space Model (VSM) for semantic composition of adjective-noun phrases. To the best of our knowledge, this is one of the first such studies of this kind. Through experiments on multiple real world benchmark datasets, we demonstrated the benefit of combining topic- and type-based features in a VSM. Additionally, we introduced a novel task of predicting vector representations of composed phrases from the brain activity of human subjects reading those phrases. We exploited a large syntactically parsed corpus to build our VSM models, and make them publicly available. We hope that the findings and resources from this paper will serve to inform future work on VSMs and semantic composition.

Acknowledgment

We are thankful to the anonymous reviewers for their constructive comments. We thank CMUs Parallel Data Laboratory (PDL) for making the OpenCloud cluster available, Justin Betteridge (CMU) for his help with parsing the corpus, and Yahoo! for providing the M45 cluster. This research has been supported in part by DARPA (under contract number FA8750-13-2-0005), NIH (NICHD award 1R01HD075328-01), Keck Foundation (DT123107), NSF (IIS0835797), and Google. Any opinions, findings, conclusions and recommendations expressed in this paper are the authors and do not necessarily reflect those of the sponsors.

[3] No pun intended.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, and Marius Pas. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. *Proceedings of NAACL-HLT 2009*.

Abdulrahman Almuhareb and Massimo Poesio. 2004. Attribute-based and value-based clustering: An evaluation. In *Proceedings of EMNLP*, pages 158–165.

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

Marco Baroni and Roberto Zamparelli. 2010. 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.

W F Battig and W E Montague. 1969. Category Norms for Verbal Items in 56 Categories: A Replication and Extension of the Connecticut Category Norms. *Journal of Experimental Psychology Monographs*, 80(3):1–46.

William Blacoe and Mirella Lapata. 2012. A Comparison of Vector-based Representations for Semantic Composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556, Jeju Island, Korea.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022.

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

Kai-min Chang, Vladimir L. Cherkassky, Tom M Mitchell, and Marcel Adam Just. 2009. Quantitative modeling of the neural representation of adjective-noun phrases to account for fMRI activation. In *Proceedings of the Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pages 638–646.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: the concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Thomas L Griffiths, Mark Steyvers, David M Blei, and Joshua B Tenenbaum. 2005. Integrating topics and syntax. *Advances in neural information processing systems*, 17.

Joachim Gross, Sylvain Baillet, Gareth R. Barnes, Richard N. Henson, Arjan Hillebrand, Ole Jensen, Karim Jerbi, Vladimir Litvak, Burkhard Maess, Robert Oostenveld, Lauri Parkkonen, Jason R. Taylor, Virginie van Wassenhove, Michael Wibral, and Jan-Mathijs Schoffelen. 2012. Good-practice for conducting and reporting MEG research. *NeuroImage*, October.

J Hall, J Nilsson, J Nivre, G Eryigit, B Megyesi, M Nilsson, and M Saers. 2007. Single Malt or Blended? A Study in Multilingual Parser Optimization. In *Proceedings of the CoNLL Shared Task Session of EMNLPCoNLL 2007*, volume s. 19-33, pages 933–939. Association for Computational Linguistics.

Nancy Ide and Keith Suderman. 2006. The American National Corpus First Release. *Proceedings of the Fifth Language Resources and Evaluation Conference (LREC)*.

Eric Jones, Travis Oliphant, Pearu Peterson, and others. 2001. SciPy: Open source scientific tools for Python.

George Karypis. 2003. CLUTO: A Clustering Toolkit. Technical Report 02-017, Department of Computer Science, University of Minnesota.

T Landauer and S Dumais. 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.

R B Lehoucq, D C Sorensen, and C Yang. 1998. *Arpack users' guide: Solution of large scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM.

Dekang Lin and Xiaoyun Wu. 2009. Phrase clustering for discriminative learning. In *Proceedings of the ACL*.

Dekang Lin. 1998. Automatic Retrieval and Clustering of Similar Words. In *COLING-ACL*, pages 768–774.

K Lund and C Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers*, 28:203–208.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–429, November.

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012a. Comparing Abstract and Concrete Conceptual Representations using Neurosemantic Decoding. In *NAACL Workshop on Cognitive Modelling and Computational Linguistics*.

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012b. Selecting Corpus-Semantic Models for Neurolinguistic Decoding. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 114–123, Montreal, Quebec, Canada.

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

S Padó and M Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Mark Palatucci, Geoffrey Hinton, Dean Pomerleau, and Tom M Mitchell. 2009. Zero-Shot Learning with Semantic Output Codes. *Advances in Neural Information Processing Systems*, 22:1410–1418.

James Pustejovsky. 1995. *The Generative Lexicon*. MIT Press, Cambridge.

Reinhard Rapp. 2003. Word Sense Discovery Based on Sense Descriptor Dissimilarity. *Proceedings of the Ninth Machine Translation Summit*, pp:315–322.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, October.

Mehrnoosh Sadrzadeh and Edward Grefenstette. 2011. A Compositional Distributional Semantics Two Concrete Constructions and some Experimental Evaluations. *Lecture Notes in Computer Science*, 7052:35–47.

Magnus Sahlgren. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Dissertation, Stockholm University.

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

Gustavo Sudre, Dean Pomerleau, Mark Palatucci, Leila Wehbe, Alona Fyshe, Riitta Salmelin, and Tom Mitchell. 2012. Tracking Neural Coding of Perceptual and Semantic Features of Concrete Nouns. *NeuroImage*, 62(1):463–451, May.

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