Distributional Composition using Higher-Order Dependency Vectors

Julie Weeds, David Weir and **Jeremy Reffin**
Department of Informatics
University of Sussex
Brighton, BN1 9QH, UK
{J.E.Weeds, D.J.Weir, J.P.Reffin}@sussex.ac.uk

Abstract

This paper concerns how to apply compositional methods to vectors based on grammatical dependency relation vectors. We demonstrate the potential of a novel approach which uses higher-order grammatical dependency relations as features. We apply the approach to adjective-noun compounds with promising results in the prediction of the vectors for (held-out) observed phrases.

1 Introduction

Vector space models of semantics characterise the meaning of a word in terms of distributional features derived from word co-occurrences. The most widely adopted basis for word co-occurrence is proximity, i.e. that two words (or more generally lexemes) are taken to co-occur when they occur together within a certain sized window, or within the same sentence, paragraph, or document. Lin (1998), in contrast, took the syntactic relationship between co-occurring words into account: the distributional features of a word are based on the word's grammatical dependents as found in a dependency parsed corpus. For example, observing that the word *glass* appears as the indirect object of the verb *fill*, provides evidence that the word *glass* has the distributional feature $\overline{\text{iobj}}$:*fill*, where $\overline{\text{iobj}}$ denotes the inverse indirect object grammatical relation. The use of grammatical dependents as word features has been exploited in the discovery of tight semantic relations, such as synonymy and hypernymy, where an evaluation against a gold standard such as WordNet (Fellbaum, 1998) can be made (Lin, 1998; Weeds and Weir, 2003; Curran, 2004).

Pado and Lapata (2007) took this further by considering not just *direct* grammatical dependents, but also including indirect dependents. Thus, observing the sentence *She filled her glass slowly* would provide evidence that the word *glass* has the distributional feature $\overline{\text{iobj}}$:advmod:*slowly* where $\overline{\text{iobj}}$:advmod captures the indirect dependency relationship between *glass* and *slowly* in the sentence.

Note that Pado and Lapata (2007) included a basis mapping function that gave their framework flexibility as to how to map paths such as $\overline{\text{iobj}}$:advmod:*slowly* onto the basis of the vector space. Indeed, the instantiation of their framework that they adopt in their experiments uses a basis mapping function that removes the dependency path to leave just the word, so $\overline{\text{iobj}}$:advmod:*slowly* would be mapped to *slowly*.

In this paper, we are concerned with the problem of distributional semantic composition. We show that the idea that the distributional semantics of a word can be captured with higher-order dependency relationships, provides the basis for a simple approach to compositional distributional semantics. While our approach is quite general, dealing with arbitrarily high-order dependency relationships, and the composition of arbitrary phrases, in this paper we consider only first and second order dependency relations, and adjective-noun composition.

In Section 2, we illustrate our proposal by showing how second order dependency relations can play a role in computing the semantics of adjective-noun composition. In Section 3 we describe a number of experiments that are intended to evaluate the approach, with the results presented in Section 4.

The basis for our evaluation follows Baroni and

Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) @ EACL 2014, pages 11–20,
Gothenburg, Sweden, April 26-30 2014.

Zamparelli (2010) and Guevara (2010). Typically, compositional distributional semantic models can be used to generate an (inferred) distributional vector for a phrase from the (observed) distributional vectors of the phrase's constituents. One of the motivations for doing this is that the observed distributional vectors for most phrases tend to be very sparse, a consequence of the frequency with which typical phrases occur in even large corpora. However, there are phrases that occur sufficiently frequently that a reasonable characterisation of their meaning *can* be captured with their observed distributional vector. Such phrases can be exploited in order to assess the quality of a model of composition. This is achieved by measuring the distributional similarity of the observed and inferred distributional vectors for these high frequency phrases.

The contributions of this paper are as follows. We propose a novel approach to phrasal composition which uses higher order grammatical dependency relations as features. We demonstrate its potential in the context of adjective-noun composition by comparing (held-out) observed and inferred phrasal vectors. Further, we compare different vector operations, different feature association scores and investigate the effect of weighting features before or after composition.

2 Composition with Higher-order Dependencies

Consider the problem of adjective-noun composition. For example, what is the meaning of the phrase *small child*? How does it relate to the meanings of the lexemes *small* and *child*? Figure 1 shows a dependency analysis for the sentence *The very small wet child cried loudly.* Tables 1 and 2 show the grammatical dependencies (with other open-class words) for the lexemes *small* and *child* which would be extracted from it.

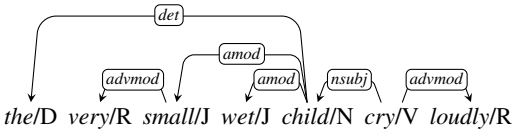

Figure 1: Example Dependency Tree

From Table 1 we see what kinds of (higher-order) dependency paths appear in the distributional features of adjectives such as *small*. Similarly, Table 2 indicates this for nouns such as *child*.

1st-order	advmod:*very*/R $\overline{\text{amod}}$:*child*
2nd-order	$\overline{\text{amod}}$:amod:*wet*/J $\overline{\text{amod}}$:$\overline{\text{nsubj}}$:*cry*/V
3rd-order	$\overline{\text{amod}}$:$\overline{\text{nsubj}}$:advmod:*loudly*/R

Table 1: Grammatical Dependencies of *small*

1st-order	amod:*wet*/J amod:*small*/J $\overline{\text{nsubj}}$:*cry*/V
2nd-order	amod:advmod:*very*/R $\overline{\text{nsubj}}$:advmod:*loudly*/R

Table 2: Grammatical Dependencies of *child*

It is clear that with a conventional grammatical dependency-based approach where only first order dependencies for *small* and *child* would be considered, there will be very little overlap between the features of nouns and adjectives because quite different grammatical relations are used in the two types of vectors, and correspondingly lexemes with different parts of speech appear at the end of these paths.

However, as our example illustrates, it is possible to align the 2nd-order feature space of adjectives with the 1st-order feature space of nouns. In this example, we have evidence that *children cry* and that *small* things *cry*. Consequently, in order to compose an adjective with a noun, we would want to align 2nd-order features of the adjective with 1st-order features of the noun; this gives us a prediction of the first order features of the noun in the context of the adjective[1].

This idea extends in a straightforward way beyond adjective-noun composition. For example, it is possible to align the 3rd order features of adjectives with 2nd order features of nouns, which is something that would be useful if one wanted to compose verbs with their arguments. These arguments will include adjective-noun compounds and therefore adjective-noun compounds require 2nd-order features which can be aligned with the first order features of the verbs. This is, however, not

[1]Note that it would also be possible to align 2nd-order features of the noun with 1st-order features of the adjective, resulting in a prediction of the first order features of the adjective in the context of the noun.

something that we will pursue further in this paper.

We now clarify how features vectors are aligned and then composed. Suppose that the lexemes w_1 and w_2 which we wish to compose are connected by relation r. Let w_1 be the head of the relation and w_2 be the dependent. In our example, w_1 is *child*, w_2 is *small* and r is amod. We first produce a reduced vector for w_2 which is designed to lie in a comparable feature space as the vector for w_1. To do this we take the set of 2nd order features of w_2 which start with the relation $\bar{r}$ and reduce them to first order features (by removing the $\bar{r}$ at the start of the path). So in our example, we create a reduced vector for *small* where features $\overline{\text{amod}}$:$\overline{\text{nsubj}}$:x for some token x are reduced to $\overline{\text{nsubj}}$:x, features $\overline{\text{amod}}$:amod:x for some token x are reduced to the feature amod:x, and features $\overline{\text{amod}}$:$\overline{\text{nsubj}}$:advmod:x for some token x are reduced to $\overline{\text{nsubj}}$:advmod:x. Once the vector for w_2 has been reduced, it can be composed with the vector for w_1 using standard vector operations.

In Section 3 we describe experiments that explore the effectiveness of this approach to distributional composition by measuring the similarity of composed vectors with observed vectors for a set of frequently occurring adjective-noun pairs (details given below). We evaluate a number of instantiations of our approach, and in particular, there are three aspects of the model where alternative solutions are available: the choice of which vector composition operation to use; the choice of how to weight dependency features; and the question as to whether feature weighting should take place before or after composition.

Vector composition operation. We consider each of the following seven alternatives: pointwise addition (`add`), pointwise multiplication (`mult`), pointwise geometric mean[2] (`gm`), pointwise maximum (`max`), pointwise minimum (`min`), first argument (`hd`), second argument (`dp`). The latter two operations simply return the first (respectively second) of the input vectors.

Feature weighting. We consider three options. Much work in this area has used positive pointwise mutual information (PPMI) (Church and Hanks, 1989) to weight the features. However, PPMI is known to over-emphasise low frequency events, and as a result there has been a recent shift towards using positive localised mutual information (PLMI) (Scheible et al., 2013) and positive normalised point wise mutual information (PNPMI) (Bouma, 2009). For definitions, see Table 3.

$$\text{PPMI}(x,y) = \begin{cases} I(x,y) & \text{if } I(x,y) > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $I(x,y) = log\frac{P(x,y)}{P(x).P(y)}$

$$\text{PLMI}(x,y) = \begin{cases} L(x,y) & \text{if } L(x,y) > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $L(x,y) = P(x,y).log(\frac{P(x,y)}{P(x).P(y)}$

$$\text{PNPMI}(x,y) = \begin{cases} N(x,y) & \text{if } N(x,y) > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $N(x,y) = \frac{1}{-log(P(y)}.log\frac{P(x,y)}{P(x).P(y)}$

Table 3: Feature Association Scores

Timing of feature weighting. We consider two alternatives: we can weight features before composition so that the composition operation is applied to weighted vectors, or we can compose vectors prior to feature weighting, in which case the composition operation is applied to unweighted vectors, and feature weighting is applied in the context of making a similarity calculation. In other work, the former order is often implied. For example, Boleda et al. (2013) state that they use "PMI to weight the co-occurrence matrix". However, if we allow the second order, features which might have a zero association score in the context of the the individual lexemes, could be considered significant in the context of the phrase.

3 Evaluation

Our experimental evaluation of the approach is based on the assumption, which is commonly made elsewhere, that where there is a reasonable amount of corpus data available for a phrase, this will generate a good estimate of the vector of the phrase. It has been shown (Turney, 2012; Baroni and Zamparelli, 2010) that such "observed" vectors are indeed reasonable for adjective-noun and noun-noun compounds. Hence, in order to evaluate the compositional models under consideration here, we compare observed phrasal vectors with inferred phrasal vectors, where the comparison is made using the cosine measure. We note that it is

[2]The geometric mean of x and y is $\sqrt{(x \cdot y)}$.

not possible to draw conclusions from the absolute value of the cosine score since this would favour models which always assign higher cosine scores. Hence, we draw conclusions from the change in cosine score with respect to a baseline within the same model.

Methodology

For each noun and adjective which occur more than a threshold number of times in a corpus, we first extract conventional first order dependency vectors. The features of these lexemes define the semantic space, and feature probabilities (for use in association scores) are calculated from this data.

Given a list of adjective-noun phrases, we extract first order vectors for the nouns and second order vectors for the adjectives, which we refer to as observed constituent vectors. We also extract first order vectors for the nouns in the context of the adjective, which we refer to as the observed phrasal vector.

For each adjective-noun pair, we build bespoke constituent vectors for the adjective and noun, in which we remove all counts which arise from co-occurrences with that specific adjective-noun pair. It is these constituent vectors that are used as the basis for inferring the vector for that particular adjective-noun phrase.

Our rationale for this is as follows. Without this modification, the observed constituent vectors will contain co-occurrences which are due to the observed adjective-noun vector co-occurrences. To see why this is undesirable, suppose that one of the adjective-noun phrases was *small child*. We take the observed vector for *small child* to be what we are calling the observed phrasal vector for *child* (in the context of *small*). Suppose that when building the observed phrasal vector, we observe the phrase *the small child cried*. This will lead to a count for the feature $\overline{\text{nsubj}}$:*cry* in the observed phrasal vector for *child*.

But if we are not careful, this same phrase will contribute to counts in the *constituent* vectors for *small* and *child*, producing counts for the features $\overline{\text{amod}}$:$\overline{\text{nsubj}}$:*cry* and $\overline{\text{nsubj}}$:*cry*, in their respective vectors. To see why these counts should not be included when building the constituent vectors that we compose to produce inferred vectors for the adjective-noun phrase *small child*, consider the case where all of the evidence for *small* things being things that can *cry* and *children* being things that can *crying* comes from having observed the phrase *small children crying*. Despite not having learnt anything about the composition of *small* and *child* in general, we would be able to infer the *cry* feature for the phrase. An adequate model of composition should be able to infer this on the basis that other *small* things have been seen to *cry*, and that non-*small children* have been seen to *cry*.

Here, we compare the proposed approach, based on higher order dependencies, with the standard method of composing conventional first-order dependency vectors. The vector operation, `hd` provides a baseline for comparison which is the same in both approaches. This baseline corresponds to a composition model where the first order dependencies of the phrase (i.e. the noun in the context of the adjective) are taken to be the same as the first order dependencies of the uncontextualized noun. For example, if we have never seen the phrase *small child* before, we would assume that it means the same as the head word *child*.

We hypothesise that it is not possible to improve on this baseline using traditional first-order dependency relation vectors, since the vector for the modifier does not contain features of the right type, but that with the proposed approach, the inferred vector for a phrase such as *small child* will be closer than observed vector for *child* to the observed vector for *small child*. We also ask the related question of whether our inferred vector for *small child* is closer than the constituent vector for *small* to the observed vector for *small child*. This comparison is achieved through use of the vector operation `dp` that ignores the vector for the head, simply returning a first-order vector derived from the dependent.

Experimental Settings

Our corpus is a mid-2011 dump of WikiPedia. This has been part-of-speech tagged, lemmatised and dependency parsed using the Malt Parser (Nivre, 2004). All major grammatical dependency relations involving open class parts of speech (nsubj, dobj, iobj, conj, amod, advmod, nnmod) have been extracted for all POS-tagged and lemmatised nouns and adjectives occurring 100 or more times. In past work with conventional dependency relation vectors we found that using a feature threshold of 100, weighting features with PPMI and a cosine similarity score work well.

For experimental purposes, we have taken

spanish	british	african	japanese
modern	classical	female	natural
digital	military	medical	musical
scientific	free	black	white
heavy	common	small	large
strong	short	long	good
similar	previous	future	original
former	subsequent	next	possible

Table 4: Adjectives considered

32 of the most frequently occurring adjectives (see Table 4). These adjectives include ones which would generally be considered intersective (e.g., *female*), subsective (e.g,, *long*) and non-subsective/intensional (e.g., *former*) (Pustejovsky, 2013) . For all of these adjectives there are at least 100 adjective-noun phrases which occur at least 100 times in the corpus. We randomly selected 50 of the phrases for each adjective. Note that our proposed method does not require any hyper parameters to be set during training, nor does it require a certain number of phrases per adjective. For the purpose of these experiments we have a list of 1600 adjective-noun phrases, all of which occur at least 100 times in WikiPedia.

4 Results and Discussion

Tables 5 and 6 summarise the average cosines for the proposed higher-order dependency approach and the conventional first-order dependency approach, respectively. In each case, we consider each combination of vector operation, feature association score, and composition timing (i.e. before, or after, vector weighting).

Table 7 shows the average improvement over the baseline (`hd`), for each combination of experimental variables, when considering the proposed higher-order dependency approach. Note that this is an average of paired differences (and not the difference of the averages in Table 6). For brevity, we omit the results for PNPMI here, since there do not appear to be substantial differences between using PPMI and PNPMI. To indicate statistical significance, we show estimated standard errors in the means. All differences are statistically significant (under a paired t-test) except those marked †.

From Table 5, we see that none of the compositional operations on conventional dependency vectors are able to beat the baseline of selecting the head vector (`hd`). This is independent of the choice of association measure and the order in which weighting and composition are carried out.

For the higher order dependency vectors (Tables 6 and 7), we note, in contrast, that some compositional operations produce large increases in cosine score compared to the head vector alone (`hd`). Table 7 examines the statistical significance of these differences. We find that for the intersective composition operations (`mult`, `min`, and `gm`), performance is statistically superior to using the head alone in all experimental conditions studied. By contrast, additive measures (`add`, `max`) typically have no impact, or decrease performance marginally relative to the head alone. An explanation for these significant differences is that intersective vector operations are able to encapsulate the way that an adjective disambiguates and specialises the sense of the noun that it is modifying.

We also note that the alternative baseline, `dp`, which estimates the features of a phrase to be the aggregation of all things which are modified by the adjective, performs significantly worse than the standard baseline, `hd`, which estimates the features of a phrase to be the features of the head noun. This is consistent with the intuition that the distributional vector for *small child* should more similar to the vector for *child* than it is to the vector for the things that can be *small*.

Considering the different intersective operations, `mult` appears to be the best choice when the feature association score is PPMI or PNPMI and `gm` appears to be the best choice when the feature association score is PLMI.

Further, PLMI consistently gives all of the vector pairings higher cosine scores than PPMI. Since PLMI assigns less weight to low frequency event and more weight to high frequency events, this suggests that all of the composition methods, including the baseline (`hd`), do better at predicting the high frequency co-occurrences. This is not surprising as these will more likely have been seen with the phrasal constituents in other contexts.

Our final observation, based on Table 6, is that the best order in which to carry out weighting and composition appears to depend on the choice of feature association score. In general, it appears better to weight the features and then compose vectors. This is always true when using PNPMI or PLMI. However, using PPMI, the highest performance is achieved by composing the raw vectors using multiplication and then weighing the

	weight:compose						compose:weight					
	PPMI		PNPMI		PLMI		PPMI		PNPMI		PLMI	
	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s
add	0.12	(0.06)	0.13	(0.05)	0.15	(0.16)	0.11	(0.05)	0.12	(0.06)	0.22	(0.20)
max	0.12	(0.06)	0.13	(0.05)	0.15	(0.16)	0.11	(0.05)	0.12	(0.06)	0.22	(0.20)
mult	0.06	(0.05)	0.06	(0.06)	0.06	(0.11)	0.07	(0.05)	0.07	(0.12)	0.07	(0.05)
min	0.05	(0.05)	0.06	(0.05)	0.04	(0.09)	0.05	(0.04)	0.05	(0.04)	0.04	(0.08)
gm	0.06	(0.05)	0.06	(0.05)	0.07	(0.11)	0.05	(0.04)	0.06	(0.04)	0.08	(0.11)
hd	**0.13**	(0.07)	**0.15**	(0.07)	**0.28**	(0.22)	**0.13**	(0.07)	**0.15**	(0.07)	**0.28**	(0.22)

Table 5: Means and Standard Deviations for Cosines Between Observed and Predicted Vectors for Conventional First-Order Dependency Based Approach.

	weight:compose						compose:weight					
	PPMI		PNPMI		PLMI		PPMI		PNPMI		PLMI	
	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s
add	0.14	(0.06)	0.16	(0.06)	0.29	(0.21)	0.10	(0.04)	0.12	(0.05)	0.29	(0.22)
max	0.10	(0.04)	0.11	(0.04)	0.27	(0.21)	0.10	(0.04)	0.11	(0.04)	0.26	(0.21)
mult	**0.30**	(0.12)	**0.33**	(0.12)	0.40	(0.29)	**0.34**	(0.10)	**0.32**	(0.10)	0.32	(0.27)
min	0.26	(0.11)	0.27	(0.11)	0.40	(0.24)	0.24	(0.10)	0.25	(0.10)	0.37	(0.23)
gm	0.27	(0.11)	0.29	(0.11)	**0.46**	(0.20)	0.26	(0.10)	0.27	(0.10)	**0.44**	(0.22)
dp	0.10	(0.05)	0.10	(0.05)	0.20	(0.20)	0.10	(0.05)	0.10	(0.05)	0.20	(0.20)
hd	0.13	(0.07)	0.15	(0.07)	0.28	(0.22)	0.13	(0.07)	0.15	(0.07)	0.28	(0.22)

Table 6: Means and Standard Deviations for Cosines Between Observed and Predicted Vectors for Proposed Higher-Order Dependency Based Approach

remaining features. This can be explained by considering the recall and precision of the composed vector's prediction of the observed vector. If we compose using gm before weighting vectors, we increase the recall of the prediction, but decrease precision. Whether we use PPMI, PNPMI or PLMI, recall of features increases from 88.8% to 99.5% and precision drops from 5.5% to 4.8%. If we compose using mult before weighting vectors, contrary to expectation, recall decreases and precision increases. Whether we use PPMI, PNPMI or PLMI, recall of features decreases from 88.8% to 59.4% but precision increases from 5.5% to 18.9%. Hence, multiplication of the raw vectors is causing a lot of potential shared features to be "lost" when the weighting is subsequently carried out (since multiplication stretches out the value space). This leads to an increase in cosines when PPMI is used for weighting, and a decrease in cosines when PLMI is used. Hence, it appears that the features being removed by multiplying the raw vectors before weighting must be low frequency co-occurrences, which are not observed with the phrase.

5 Related Work

In this work, we bring together ideas from several different strands of distributional semantics: incorporating syntactic information into the distributional representation of a lexeme; representing phrasal meaning by creating distributional representations through composition; and representing word meaning in context by modifying the distributional representation of a word.

The use of syntactic structure in distributional representations is not new. Two of the earliest proponents of distributional semantics, Lin (1998) and Lee (1999) used features based on first order dependency relations between words in their distributional representations. More recently, Pado and Lapata (2007) propose a semantic space based on dependency paths. This model outperformed traditional word-based models which do not take syntax into account in a synonymy relation detection task and a prevalent sense acquisition task.

The problem of representing phrasal meaning has traditionally been tackled by taking vector representations for words (Turney and Pantel, 2010) and combining them using some function to pro-

	weight:compose				compose:weight			
	PPMI		PLMI		PPMI		PLMI	
	$\bar{x}$	$s_{\bar{x}}$	$\bar{x}$	$s_{\bar{x}}$	$\bar{x}$	$s_{\bar{x}}$	$\bar{x}$	$s_{\bar{x}}$
add	0.01	(0.001)	†0.004	(0.003)	-0.03	(0.001)	†0.006	(0.004)
max	-0.03	(0.001)	-0.01	(0.003)	-0.04	(0.001)	-0.02	(0.003)
mult	**0.16**	(0.002)	0.11	(0.006)	**0.21**	(0.002)	0.03	(0.006)
min	0.13	(0.001)	0.11	(0.007)	0.10	(0.001)	0.09	(0.007)
gm	0.14	(0.001)	**0.18**	(0.005)	0.12	(0.001)	**0.16**	(0.005)
dp	-0.03	(0.002)	-0.09	(0.007)	-0.04	(0.002)	-0.09	(0.007)

Table 7: Means and Standard Errors for Increases in Cosine with respect to the hd Baseline for Proposed Higher-Order Dependency Based Approach. All differences statistically significant (under a paired t-test) except those marked †.

duce a data structure that represents the phrase or sentence. Mitchell and Lapata (2008, 2010) found that simple additive and multiplicative functions applied to proximity-based vector representations were no less effective than more complex functions when performance was assessed against human similarity judgements of simple paired phrases.

The simple functions evaluated by Mitchell and Lapata (2008) are generally acknowledged to have serious theoretical limitations in their treatment of composition. How can a commutative function such as multiplication or addition provide different interpretations for different word orderings such as *window glass* and *glass window*? The majority of attempts to rectify this have offered a more complex, non-commutative function — such as weighted addition — or taken the view that some or all words are no longer simple vectors. For example, in the work of Baroni and Zamparelli (2010) and Guevara (2010), an adjective is viewed as a modifying function and represented by a matrix. Coecke et al. (2011) and Grefenstette et al. (2013) also incorporate the notion of function application from formal semantics. They derived function application from syntactic structure, representing functions as tensors and arguments as vectors. The MV-RNN model of Socher et al. (2012) broadened the Baroni and Zamparelli (2010) approach; all words, regardless of part-of-speech, were modelled with both a vector and a matrix. This approach also shared features with Coecke et al. (2011) in using syntax to guide the order of phrasal composition. These higher order structures are typically learnt or induced using a supervised machine learning technique. For example, Baroni and Zamparelli (2010) learnt their adjectival matrixes by performing regression analysis over pairs of observed nouns and adjective-noun phrases. As a consequence of the computational expense of the machine learning techniques involved, implementations of these approaches typically require a considerable amount of dimensionality reduction.

A long-standing topic in distributional semantics has been the modification of a canonical representation of a lexeme's meaning to reflect the context in which it is found. Typically, a canonical vector for a lexeme is estimated from all corpus occurrences and the vector then modified to reflect the instance context (Lund and Burgess, 1996; Erk and Padó, 2008; Mitchell and Lapata, 2008; Thater et al., 2009; Thater et al., 2010; Thater et al., 2011; Van de Cruys et al., 2011; Erk, 2012). As described in Mitchell and Lapata (2008, 2010), lexeme vectors have typically been modified using simple additive and multiplicative compositional functions. Other approaches, however, share with our proposal the use of syntax to drive modification of the distributional representation (Erk and Padó, 2008; Thater et al., 2009; Thater et al., 2010; Thater et al., 2011). For example, in the SVS representation of Erk and Padó (2008), a word was represented by a set of vectors: one which encodes its lexical meaning in terms of distributionally similar words[3], and one which encodes the selectional preferences of each grammatical relation it supports. A word's meaning vector was updated in the context of another word by combining it with the appropriate selectional preferences vec-

[3]These are referred to as second-order vectors using the terminology of Grefenstette (1994) and Schütze (1998). However, this refers to a second-order affinity between the words and is not related to the use of grammatical dependency relations.

tor of the contextualising word.

Turney (2012) offered a model of phrasal level similarity which combines assessments of word-level semantic relations. This work used two different word-level distributional representations to encapsulate two types of similarity. Distributional similarity calculated from proximity-based features was used to estimate domain similarity and distributional similarity calculated from syntactic pattern based features is used to estimate functional similarity. The similarity of a pair of compound noun phrases was computed as a function of the similarities of the components. Crucially different from other models of phrasal level similarity, it does not attempt to derive modified vectors for phrases or words in context.

6 Conclusions and Further Work

Vectors based on grammatical dependency relations are known to be useful in the discovery of tight semantic relations, such as synonymy and hypernymy, between lexemes (Lin, 1998; Weeds and Weir, 2003; Curran, 2004). It would be useful to be able to extend these methods to determine similarity between phrases (of potentially different lengths). However, conventional approaches to composition, which have been applied to proximity-based vectors, cannot sensibly be used on vectors that are based on grammatical dependency relations.

In our approach, we consider the vector for a phrase to be the vector for the head lexeme in the context of the other phrasal constituents. Like Pado and Lapata (2007), we extend the concept of a grammatical dependency relation feature to include dependency relation paths which incorporate higher-order dependencies between words. We have shown how it is possible to align the dependency path features for words of different syntactic types, and thus produce composed vectors which predict the features of one constituent in the context of the other constituent.

In our experiments with AN compounds, we have shown that these predicted vectors are closer than the head constituent's vector to the observed phrasal vector. We have shown this is true even when the observed phrase is in fact unobserved, i.e. when its co-occurrences do not contribute to the constituents' vectors. Consistent with work using proximity-based vectors, we have found that intersective operations perform substantially better than additive operations. This can be understood by viewing the intersective operations as encapsulating the way that adjectives can specialise the meaning of the nouns that they modify.

We have investigated the interaction between the vector operation used for composition, the feature association score and the timing of applying feature weights. We have found that multiplication works best if using PPMI to weight features, but that geometric mean is better if using the increasingly popular PLMI weighting measure. Whilst applying an intersective composition operation before applying feature weighting does allow more features to be retained in the predicted vector (it is possible to achieve 99.5% recall), in general, this does not correspond with an increase in cosine scores. In general, the corresponding drop in precision (i.e., the over-prediction of unobserved features) causes the cosine to decrease. The one exception to this is using multiplication with the PPMI feature weighting score. Here we actually see a drop in recall, and an increase in precision due to the nature of multiplication and PPMI.

One assumption that has been made throughout the work, is that the observed phrasal vector provides a good estimate of the distributional representation of the phrase and, consequently, the best composition method is the one which returns the most similar prediction. However, in general, we notice that while the recall of the compositional methods is good, the precision is very low. Lack of precision may be due to the prevalence of plausible, but unobserved, co-occurrences of the phrase. Consequently, this introduces uncertainty into the conclusions which can be drawn from a study such as this. Further work is required to develop effective intrinsic and extrinsic evaluations of models of composition.

A further interesting area of study is whether distributional models that include higher-order grammatical dependencies can tell us more about the lexical semantics of a word than the conventional first-order models, for example by distinguishing semantic relations such as synonymy, antonymy, hypernymy and co-hyponymy.

Acknowledgements

This work was funded by UK EPSRC project EP/IO37458/1 "A Unified Model of Compositional and Distributional Compositional Semantics: Theory and Applications".

References

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*.

Gemma Boleda, Marco Baroni, The Nghia Pham, and Louise McNally. 2013. Intensionality was only alleged: On adjective-noun composition in distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 35–46, Potsdam, Germany, March. Association for Computational Linguistics.

Gerlof Bouma. 2009. Normalised (point wise) mutual information in collocation extraction, from form to meaning: Processing texts automatically. In *Proceedings of the Biennial International Conference of the German Society for Computational Linguistics and Language Technology*.

Kenneth Ward Church and Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Meeting on Association for Computational Linguistics*, ACL '89, pages 76–83, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2011. Mathematical foundations for a compositional distributed model of meaning. *Linguistic Analysis*, 36(1-4):345–384.

James Curran. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 897–906, Honolulu, Hawaii, October. Association for Computational Linguistics.

Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: a survey. *Language and Linguistics Compass*, 6(10):635–653.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Edward Grefenstette, Georgiana Dinu, Yao-Zhong Zhang, Mehrnoosh Sadrzadeh, and Marco Baroni. 2013. Multi-step regression learning for compositional distributional semantics. *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*.

Gregory Grefenstette. 1994. Corpus-derived first, second and third-order word affinities. In *Proceedings of Euralex 1994*.

Emiliano Guevara. 2010. A Regression Model of Adjective-Noun Compositionality in Distributional Semantics. In *Proceedings of the ACL GEMS Workshop*, pages 33–37.

Lillian Lee. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, College Park, Maryland, USA, June. Association for Computational Linguistics.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistings (COLING 1998)*.

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

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio, June. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the ACL Workshop on Incremental Parsing*, pages 50–57.

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

James Pustejovsky. 2013. Inference patterns with intensional adjectives. In *Proceedings of the IWCS Workshop on Interoperable Semantic Annotation*, Potsdam,Germany, March. Association for Computational Linguistics.

Silke Scheible, Sabine Schulte im Walde, and Sylvia Springorum. 2013. Uncovering distributional differences between synonyms and antonyms in a word space model. In *Proceedings of the International Joint Conference on Natural Language Processing*, pages 489–497, Nagoya, Japan.

Heinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics.

Stefan Thater, Georgiana Dinu, and Manfred Pinkal. 2009. Ranking paraphrases in context. In *Proceedings of the 2009 Workshop on Applied Textual Inference*, pages 44–47, Suntec, Singapore, August. Association for Computational Linguistics.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 948–957, Uppsala, Sweden, July. Association for Computational Linguistics.

Stefan Thater, Hagen Frstenau, and Manfred Pinkal. 2011. Word meaning in context: A simple and effective vector model. In *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*.

P. D. Turney and P. Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Peter D. Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44.

Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2011. Latent vector weighting for word meaning in context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1012–1022, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 81–88, Sapporo, Japan.