Improving Sparse Word Representations with Distributional Inference for Semantic Composition

Thomas Kober, Julie Weeds, Jeremy Reffin and **David Weir**
TAG laboratory, Department of Informatics, University of Sussex
Brighton, BN1 9QH, UK
{t.kober, j.e.weeds, j.p.reffin, d.j.weir}@sussex.ac.uk

Abstract

Distributional models are derived from co-occurrences in a corpus, where only a small proportion of all possible plausible co-occurrences will be observed. This results in a very sparse vector space, requiring a mechanism for inferring missing knowledge. Most methods face this challenge in ways that render the resulting word representations uninterpretable, with the consequence that semantic composition becomes hard to model. In this paper we explore an alternative which involves explicitly inferring unobserved co-occurrences using the distributional neighbourhood. We show that distributional inference improves sparse word representations on several word similarity benchmarks and demonstrate that our model is competitive with the state-of-the-art for adjective-noun, noun-noun and verb-object compositions while being fully interpretable.

1 Introduction

The aim of distributional semantics is to derive meaning representations based on observing co-occurrences of words in large text corpora. However not all plausible co-occurrences will be observed in any given corpus, resulting in word representations that only capture a fragment of the meaning of a word. For example the verbs "walking" and "strolling" may occur in many different and possibly disjoint contexts, although both verbs would be equally plausible in numerous cases. This subsequently results in incomplete representations for both lexemes. In addition, models based on counting co-occurrences face the general problem of sparsity in a very high-dimensional vector space. The most common approaches to these challenges have involved the use of various techniques for dimensionality reduction (Bullinaria and Levy, 2012; Lapesa and Evert, 2014) or the use of low-dimensional and dense neural word embeddings (Mikolov et al., 2013; Pennington et al., 2014). The common problem in both of these approaches is that composition becomes a black-box process due to the lack of interpretability of the representations. Count-based models are therefore a very attractive line of work with regards to a number of important long-term research challenges, most notably the development of an adequate model of distributional compositional semantics. In this paper we propose the use of distributional inference (DI) to inject unobserved but plausible distributional semantic knowledge into the vector space by leveraging the intrinsic structure of the distributional neighbourhood. This results in richer word representations and furthermore mitigates the sparsity effect common in high-dimensional vector spaces, while remaining fully interpretable.

Our contributions are as follows: we show that typed and untyped sparse word representations, enriched by distributional inference, lead to performance improvements on several word similarity benchmarks, and that a higher-order dependency-typed vector space model, based on "Anchored Packed Dependency Trees (APTs)" (Weir et al., 2016), is competitive with the state-of-the-art for adjective-noun, noun-noun and verb-object compositions. Using our method, we are able to bridge the gap in performance between high dimensional interpretable mod-

els and low dimensional non-interpretable models and offer evidence to support a possible explanation of why high-dimensional models usually perform worse, together with a simple, practical method for over-coming this problem. We furthermore demonstrate that *intersective* approaches to composition benefit more from distributional inference than composition by *union* and highlight the ability of composition by *intersection* to disambiguate the meaning of a phrase in a local context.

The remainder of this paper is structured as follows: we discuss related work in section 2, followed by an introduction of the APT framework for semantic composition in section 3. We describe distributional inference in section 4 and present our experimental work, together with our results in section 5. We conclude this paper and outline future work in section 6.

2 Related Work

Our method follows the distributional smoothing approach of Dagan et al. (1994) and Dagan et al. (1997). In these works the authors are concerned with smoothing the probability estimate for unseen words in bigrams. This is achieved by measuring which unobserved bigrams are more likely than others on the basis of the Kullback-Leibler divergence between bigram distributions. This has led to significantly improved performance on a language modelling for speech recognition task, as well as for word-sense disambiguation in machine translation (Dagan et al., 1994; Dagan et al., 1997). More recently Padó et al. (2013) used a distributional approach for smoothing derivationally related words, such as *oldish – old*, as a back-off strategy in case of data sparsity. However, none of these approaches have used distributional inference as a general technique for directly enriching sparse distributional vector representations, or have explored its behaviour for semantic composition.

Compositional models of distributional semantics have become an increasingly popular topic in the research community. Starting from simple pointwise additive and multiplicative approaches to composition, such as Mitchell and Lapata (2008; 2010), and Blacoe and Lapata (2012), to tensor based models, such as Baroni and Zamparelli (2010), Coecke et al. (2010), Grefenstette et al. (2013) and Paperno et al. (2014), and neural network based approaches, such as Socher et al. (2012), Le and Zuidema (2015), Mou et al. (2015) and Tai et al. (2015). Zanzotto et al. (2015) provide a decompositional analysis of how similarity is affected by distributional composition, and link compositional models to convolution kernels. Most closely related to our approach of composition are the works of Thater et al. (2010), Thater et al. (2011) and Weeds et al. (2014), which aim to provide a general model of compositionality in a typed distributional vector space. In this paper we adopt the approach to distributional composition introduced by Weir et al. (2016), whose APT framework is based on a higher-order dependency-typed vector space, however they do not address the issue of sparsity in their work.

3 Background

Distributional vector space models can broadly be categorised into untyped proximity based models and typed models (Baroni and Lenci, 2010). Examples of the former include Deerwester et al. (1990); Lund and Burgess (1996); Curran (2004); Sahlgren (2006); Bullinaria and Levy (2007) and Turney and Pantel (2010). These models count the number of times every word in a large corpus co-occurs with other words within a specified spatial context window, without leveraging the structural information of the text. Typed models on the other hand, take the grammatical relation between two words for a co-occurrence event into account. Early proponents of that approach are Grefenstette (1994) and Lin (1998). More recent work by Padó and Lapata (2007), Erk and Padó (2008) and Weir et al. (2016) uses dependency paths to build a structured vector space model. In both kinds of models, the raw counts are usually transformed by Positive Pointwise Mutual Information (PPMI) or a variant of it (Church and Hanks, 1990; Niwa and Nitta, 1994; Scheible et al., 2013; Levy and Goldberg, 2014).

In the following we will give an explanation of the theory of composition with APTs as introduced by Weir et al. (2016), which we adopt in this paper. In addition to direct relations between two words, the APT model also considers *inverse* and *higher*

order relations. Inverse relations are denoted with a horizontal bar above the dependency relation, such as $\overline{\texttt{amod}}$ for an inverse adjectival modifier. Higher order dependencies are separated by a colon as in the second order distributional feature $\overline{\texttt{dobj}}\texttt{:nsubj}$. The example below illustrates how raw text is processed to retrieve elementary representations in our APT model. As an example we consider a lowercased corpus consisting of the sentences:

we folded the clean clothes
i like your clothes
we bought white shoes yesterday
he folded the white sheets

We dependency parse the raw sentences and, following Weir et al. (2016), align and aggregate the resulting parse trees according to their dependency type as shown in Figure 1. For example the lexeme *clothes* has the distributional features amod:*dry* and $\overline{\texttt{dobj}}\texttt{:nsubj:}$*we* among others. Over a large corpus, this results in a very high-dimensional and sparse vector space, which due to its typed nature is much sparser than for untyped models.

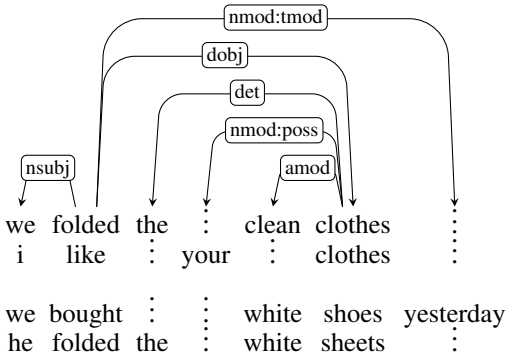

Figure 1: Aligned Packed Dependency Tree representation of the example sentences.

Composition with APTs

Composition is linguistically motivated by the principle of compositionality, which states that the meaning of a complex expression is fully determined by its structure and the meanings of its constituents (Frege, 1884). Many simple approaches to semantic composition neglect the structure and lose information in the composition process. For example, the phrases *house boat* and *boat house* have the exact same representation when composition is done via a pointwise arithmetic operation. Despite performing well in a number of studies, this commutativity is not desirable for a fine grained understanding of the semantics of natural language. When performing composition with APTs, we adopt the method introduced by Weir et al. (2016) which views distributional composition as a process of contextualisation. For composing the adjective *white* with the noun *clothes* via the dependency relation amod we need to consider how the adjective interacts with the noun in the vector space. The distributional features of white describe things that are white via their first order relations such as $\overline{\texttt{amod}}$, and things that can be done to white things, such as *bought* via $\overline{\texttt{amod}}\texttt{:}\overline{\texttt{dobj}}$ in the example above.

Table 1 shows a number of features extracted from the aligned dependency trees in Figure 1 and highlights that adjectives and nouns do not share many features if only first order dependencies would be considered. However through the inclusion of inverse and higher order dependency paths we can observe that the second order features of the adjective align with the first order features of the noun. For composition, the adjective *white* needs to be offset by its *inverse* relation to *clothes*[1] making it distributionally similar to a noun that has been modified by *white*. Offsetting can be seen as shifting the current viewpoint in the APT data structure and is necessary for aligning the feature spaces for composition (Weir et al., 2016). We are then in a position to compose the offset representation of *white* with the vector for *clothes* by the *union* or the *intersection* of their features.

Table 2 shows the resulting feature spaces of the composed vectors. It is worth noting that any arithmetic operation can be used to combine the counts of the aligned features, however for this paper we use pointwise addition for both composition functions. One of the advantages of this approach to composition is that the inherent interpretability of count-based models naturally expands beyond the word level, allowing us to study the distributional semantics of phrases in the same space as words. Due to offsetting one of the constituents, the composition operation is not commutative and hence avoids identical representations for *house boat* and *boat house*. However, the typed nature of our vector space re-

[1]The inverse of $\overline{\texttt{amod}}$ is just amod.

white			*clothes*	
Distributional Features	**Offset Features (by amod)**	**Co-occurrence Count**	**Distributional Features**	**Co-occurrence Count**
$\overline{\texttt{amod}}$:*shoes*	:*shoes*	1	amod:*clean*	1
$\overline{\texttt{amod}}$:$\overline{\texttt{dobj}}$:*bought*	$\overline{\texttt{dobj}}$:*bought*	1	$\overline{\texttt{dobj}}$:*like*	1
$\overline{\texttt{amod}}$:$\overline{\texttt{dobj}}$:*folded*	$\overline{\texttt{dobj}}$:*folded*	1	$\overline{\texttt{dobj}}$:*folded*	1
$\overline{\texttt{amod}}$:$\overline{\texttt{dobj}}$:nsubj:*we*	$\overline{\texttt{dobj}}$:nsubj:*we*	1	$\overline{\texttt{dobj}}$:nsubj:*we*	1

Table 1: Example feature spaces for the lexemes *white* and *clothes* extracted from the dependency tree of Figure 1. Not all features are displayed for space reasons. Offsetting $\overline{\texttt{amod}}$:*shoes* by amod results in an empty dependency path, leaving just the word co-occurrence :*shoes* as feature.

Composition by *union*		Composition by *intersection*	
Distributional Features	**Co-occurrence Count**	**Distributional Features**	**Co-occurrence Count**
:*shoes*	1		
amod:*clean*	1		
$\overline{\texttt{dobj}}$:*bought*	1		
$\overline{\texttt{dobj}}$:*folded*	2	$\overline{\texttt{dobj}}$:*folded*	2
$\overline{\texttt{dobj}}$:*like*	1		
$\overline{\texttt{dobj}}$:nsubj:*we*	2	$\overline{\texttt{dobj}}$:nsubj:*we*	2

Table 2: Comparison of composition by *union* and composition by *intersection*. Not all features are displayed for space reasons.

sults in extreme sparsity, for example while the untyped VSM has 130k dimensions, our APT model can have more than 3m dimensions. We therefore need to enrich the elementary vector representations with the distributional information of their nearest neighbours to ease the sparsity effect and infer missing information. Due to the syntactic nature of our composition operation it is not straightforward to apply common dimensionality reduction techniques such as SVD, as the type information needs to be preserved.

4 Distributional Inference

Following Dagan et al. (1994) and Dagan et al. (1997), we propose a simple unsupervised algorithm for enriching sparse vector representations with their nearest neighbours. We show that our distributional inference algorithm improves performance for untyped and typed models on several word similarity benchmarks, as well as being competitive with the state-of-the-art on semantic composition. As shown in algorithm 1 below, we iterate over all word vectors w in a given distributional model M, and add the vector representations of the nearest neighbours n, determined by cosine similarity, to the representation of the enriched word vector w'. The parameter α in line 4 scales the contribution of the original word vector to the resulting enriched representation. In this work we always chose α to be identical to the number of neighbours used for distributional inference. For example, if we used 10 neighbours for DI, we would set $\alpha = 10$, which we found sufficient to prevent the neighbours from dominating the vector representation. In our experiments we kept the input distributional model fixed, however it is equally possible to update the given model in an online fashion, adding some amount of stochasticity to the enriched word vector representations. There is a number of possibilities for the neighbour retrieval function $neighbours()$ and we explore several options in this paper. The algorithm furthermore is agnostic to the input distributional model, for example it is possible to use completely different vector space models for querying neighbours and enrichment.

Algorithm 1 Distributional Inference

```
1:  procedure DIST_INFERENCE(M)
2:      init M'
3:      for all w in M do
4:          w' ← w × α
5:          for all n in neighbours(M, w) do
6:              w' ← w' + n
7:              add w' to M'
8:          end for
9:      end for
10:     return M'
11: end procedure
```

Static Top n Neighbour Retrieval

The perhaps simplest way is to choose the top n most similar neighbours for each word in the vector space and enrich the respective vector representations with them.

Density based Neighbour Retrieval

This approach has its roots in kernel density estimation (Parzen, 1962), however instead of defining a static global parzen window, we set the window size for every word individually, depending on the distance to its nearest neighbour, plus a threshold. For example if the cosine distance between the target vector and its top neighbour is 0.5, we use a window size of $0.5 + \epsilon$ for that word. In our experiments we typically define ϵ to be proportional to the distance of the nearest neighbour (e.g. $\epsilon = 0.5 \times 0.1$).

WordNet based Neighbour Retrieval

Instead of leveraging the intrinsic structure of our distributional vector space, we retrieve neighbours by querying WordNet (Fellbaum, 1998), and treat synsets with agreeing PoS tags as the nearest neighbours of any target vector. This restricts the retrieved neighbours to synonyms only.

5 Experiments

Our model is based on a cleaned October 2013 Wikipedia dump, which excludes all pages with fewer than 20 page views, resulting in a corpus of approximately 0.6 billion tokens (Wilson, 2015). The corpus is lowercased, tokenised, lemmatised, PoS tagged and dependency parsed with the Stanford NLP tools, using universal dependencies (Manning et al., 2014; de Marneffe et al., 2014). We then build our APT model with first, second and third order relations. We remove distributional features with a count of less than 10, and vectors containing fewer than 50 non-zero entries. The raw counts are subsequently transformed to PPMI weights. The untyped vector space model is built from the same lowercased, tokenised and lemmatised Wikipedia corpus. We discard terms with a frequency of less than 50 and apply PPMI to the raw co-occurrence counts.

Shifted PPMI

We explore a range of different values for shifting the PPMI scores as these have a significant impact on the performance of the APT model. The effect of shifting PPMI scores for untyped vector space models has already been explored in Levy and Goldberg (2014), and Levy et al. (2015), thus we only present results for the APT model. As shown in equation 1, PMI is defined as the log of the ratio of the joint probability of observing a word w and a context c together, and the product of the respective marginals of observing them separately. In our APT model, a context c is defined as a dependency relation together with a word.

$$\begin{aligned} PMI(w,c) &= \log \frac{P(w,c)}{P(w)P(c)} \\ SPPMI(w,c) &= \max(PMI(w,c) - \log k, 0) \end{aligned} \tag{1}$$

As PMI is negatively unbounded, PPMI is used to ensure that all values are greater than or equal to 0. Shifted PPMI (SPPMI) subtracts a constant from any PMI score before applying the PPMI threshold. We experiment with values of 1, 5, 10, 40 and 100 for the shift parameter k.

5.1 Word Similarity Experiments

We first evaluate our models on 3 word similarity benchmarks, MEN (Bruni et al., 2014), which is testing for *relatedness* (e.g. meronymy or holonymy) between terms, SimLex-999 (Hill et al., 2015), which is testing for *substitutability* (e.g. synonymy, antonymy, hyponymy and hypernymy), and WordSim-353 (Finkelstein et al., 2001), where we use the version of Agirre et al. (2009), who split the dataset into a *relatedness* and a *substitutability* subset. Baroni and Lenci (2011) have shown that untyped models are typically better at capturing *relatedness*, whereas typed models are better at encoding *substitutability*. Performance is measured by computing Spearman's ρ between the cosine similarities of the vector representations and the corresponding aggregated human similarity judgements. For these experiments we keep the number of neighbours that a word vector can consume fixed at 30. This value is based on preliminary experiments on WordSim-353 (see Figure 2) using the static top n neighbour retrieval function and a PPMI shift of $k = 40$. Figure 2 shows that distributional inference improves performance for any number of neighbours over a model without DI (marked as horizontal dashed lines for each WordSim-353 subset) and peaks at a value of

APTs	MEN		SimLex-999		WordSim-353 (rel)		WordSim-353 (sub)	
	without DI	*with DI*	*without DI*	*with DI*	*without DI*	*with DI*	*without DI*	*with DI*
$k = 1$	0.54	0.52	0.31	0.30	0.34	0.27	0.62	0.60
$k = 5$	0.64	0.65	0.35	**0.36**	0.56	0.51	0.74	0.73
$k = 10$	0.63	0.66	0.35	**0.36**	0.56	0.55	0.75	0.74
$k = 40$	0.63	**0.68**	0.30	0.32	0.55	**0.61**	0.75	**0.76**
$k = 100$	0.61	0.67	0.26	0.29	0.47	0.60	0.71	0.72

Table 3: Effect of the magnitude of the shift parameter k in SPPMI on the word similarity tasks. Boldface means best performance per dateset.

30. Performance slightly degrades with more neighbours. For the untyped VSM we use a symmetric window of 5 on either side of the target word.

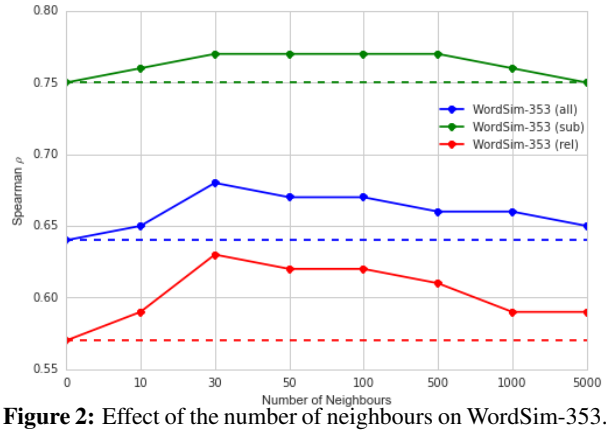

Figure 2: Effect of the number of neighbours on WordSim-353.

Table 3 highlights the effect of the SPPMI shift parameter k, while keeping the number of neighbours fixed at 30 and using the static top n neighbour retrieval function. For the APT model, a value of $k = 40$ performs best (except for SimLex-999, where smaller shifts give better results), with a performance drop-off for larger shifts. In our experiments we find that a shift of $k = 1$ results in top performance for the untyped vector space model. It appears that shifting the PPMI scores in the APT model has the effect of cleaning the vectors from noisy PPMI artefacts, which reinforces the predominant sense, while other senses get suppressed. Subsequently, this results in a cleaner neighbourhood around the word vector, dominated by a single sense. This explains why distributional inference slightly degrades performance for smaller values of k.

Table 4 shows that distributional inference successfully infers missing information for both model types, resulting in improved performance over models without the use of DI on all datasets. The improvements are typically larger for the APT model, suggesting that it is missing more distributional knowledge in its elementary representations than untyped models. The density window and static top n neighbour retrieval functions perform very similar, however the static approach is more consistent and never underperforms the baseline for either model type on any dataset. The WordNet based neighbour retrieval function performs particularly well on SimLex-999. This can be explained by the fact that antonyms, which frequently happen to be among the nearest neighbours in distributional vector spaces, are regarded as dissimilar in SimLex-999, whereas the WordNet neighbour retrieval function only returns synonyms. The results furthermore confirm the effect that untyped models perform better on datasets modelling *relatedness*, whereas typed models work better for *substitutability* tasks (Baroni and Lenci, 2011).

5.2 Composition Experiments

Our approach to semantic composition as described in section 3 requires the dimensions of our vector space models to be meaningful and interpretable. However, the problem of missing information is amplified in compositional settings as many compatible dimensions between words are not observed in the source corpus. It is therefore crucial that distributional inference is able to inject some of the missing information in order to improve the composition process. For the experiments involving semantic composition, we enrich the elementary representations of the phrase constituents before composition.

We first conduct a qualitative analysis for our APT model and observe the effect of distributional inference on the nearest neighbours of composed adjective-noun, noun-noun and verb-object compounds. In these experiments, we show how dis-

APTs ($k = 40$)	No Distributional Inference	Density Window	Static Top n	WordNet
MEN	0.63	0.67	**0.68**	0.63
SimLex-999	0.30	0.32	0.32	**0.38**
WordSim-353 (rel)	0.55	**0.62**	0.61	0.56
WordSim-353 (sub)	0.75	**0.78**	0.76	0.77
Untyped VSM ($k = 1$)	**No Distributional Inference**	**Density Window**	**Static Top n**	**WordNet**
*MEN**	**0.71**	**0.71**	**0.71**	**0.71**
SimLex-999	0.30	0.29	0.30	**0.36**
WordSim-353 (rel)	0.60	**0.64**	**0.64**	0.52
WordSim-353 (sub)	0.70	**0.73**	0.72	0.67

Table 4: Neighbour retrieval function comparison. Boldface means best performance on a dataset *per* VSM type. *) With 3 significant figures, the density window approach (0.713) is slightly better than the baseline without DI (0.708), static top n (0.710) and WordNet (0.710).

Phrase	Comp.	Union	Union (with DI)	Intersection	Intersection (with DI)
national government	AN	government, regime, ministry	government, regime*, european state*	federal assembly, government, monopoly	federal assembly, government, local office
small house	AN	house, public building, building	house, public building, large quantity*	apartment, cottage, cabin	cottage, apartment, cabin
party leader	NN	leader, market leader, government leader	leader, government leader, market leader*	party official, NDP, leader	**government leader**, party official, **opposition member**
training programme	NN	programme, action programme, television programme	programme, action programme*, television programme*	training college, trainee, education course	training college, education course, **seminar**
win battle	VO	win, win match, ties	win, win match, **fight war**	win match, win, have	**fight war**, **fight**, win match
emphasise need	VO	emphasise, underline, underscore	emphasise, underline, underscore	emphasise, prioritize, negate	emphasise, **stress importance**, **underline**

Table 5: Nearest neighbours AN, NN and VO pairs in the Mitchell and Lapata (2010) dataset, with and without distributional inference. Words and phrases marked with * denote spurious neighbours, boldfaced words and phrases mark improved neighbours.

APTs	No Distributional Inference		Density Window		Static Top n		WordNet	
	intersection	*union*	*intersection*	*union*	*intersection*	*union*	*intersection*	*union*
Adjective-Noun	0.10	<u>0.41</u>	0.31	0.39	0.25	0.40	0.12	<u>0.41</u>
Noun-Noun	0.18	0.42	0.34	0.38	0.37	<u>0.45</u>	0.24	0.36
Verb-Object	0.17	<u>0.36</u>	<u>0.36</u>	<u>0.36</u>	0.34	0.35	0.25	<u>0.36</u>
Average	0.15	**0.40**	0.34	0.38	0.32	**0.40**	0.20	0.38

Table 6: Neighbour retrieval function. Underlined means best performance per phrase type, boldface means best average performance overall.

tributional inference changes the neighbourhood in which composed phrases are embedded, and highlight the difference between composition by *union* and composition by *intersection*. For this experiment we use the static top n neighbour retrieval function with 30 neighbours and $k = 40$.

Table 5 shows a small number of example phrases together with their top 3 nearest neighbours, computed from the union of all words in the Wikipedia corpus and all phrase pairs in the Mitchell and Lapata (2010) dataset. As can be seen, nearest neighbours of phrases can be either single words or other composed phrases. Words or phrases marked with "*" in Table 5 mean that DI introduced, or failed to downrank, a spurious neighbour, while boldface means that performing distributional inference resulted in a neighbourhood more coherent with the query phrase than without DI.

Table 5 shows that composition by *union* is unable to downrank unrelated neighbours introduced by distributional inference. For example *large quantity* is incorrectly introduced as a top ranked neighbour for the phrase *small house*, due to the proximity of *small* and *large* in the vector space. The phrases *market leader* and *television programme* are two examples of incoherent neighbours, which the composition function was unable to downrank and where DI could not improve the neighbourhood. Composition by *intersection* on the other hand vastly benefits from distributional inference. Due to the increased sparsity induced by the composition process, a neighbourhood without DI produces numer-

ous spurious neighbours as in the case of the verb *have* as a neighbour for *win battle*. Distributional inference introduces qualitatively better neighbours for almost all phrases. For example, *government leader* and *opposition member* are introduced as top ranked neighbours for the phrase *party leader*, and *stress importance* and *underline* are introduced as new top neighbours for the phrase *emphasise need*. These results show that composition by *union* does not have the ability to disambiguate the meaning of a word in a given phrasal context, whereas composition by *intersection* has that ability but requires distributional inference to unleash its full potential.

For a quantitative analysis of distributional inference for semantic composition, we evaluate our model on the composition dataset of Mitchell and Lapata (2010), consisting of 108 adjective-noun, 108 noun-noun, and 108 verb-object pairs. The task is to compare the model's similarity estimates with the human judgements by computing Spearman's ρ. For comparing the performance of the different neighbour retrieval functions, we choose the same parameter settings as in the word similarity experiments ($k = 40$ and using 30 neighbours for DI).

Table 6 shows that the static top n and density window neighbour retrieval functions perform very similar again. The density window retrieval function outperforms static top n for composition by *intersection* and *vice versa* for composition by *union*. The WordNet approach is competitive for composition by *union*, but underperfoms the other approaches for composition by *intersection* significantly. For further experiments we use the static top n approach as it is computationally cheap and easy to interpret due to the fixed number of neighbours. Table 6 also shows that while composition by *intersection* is significantly improved by distributional inference, composition by *union* does not appear to benefit from it.

Composition by *Union* or *Intersection*

Both model types in this study support composition by *union* as well as composition by *intersection*. In untyped models, composition by *union* and composition by *intersection* can be achieved by pointwise addition and pointwise multiplication respectively. The major difference between composition in the APT model and the untyped model is that in the former, composition is not commutative due to offsetting the modifier in a dependency relation (see section 3). Blacoe and Lapata (2012) showed that an intersective composition function such as pointwise multiplication represents a competitive and robust approach in comparison to more sophisticated composition methods. For the final set of experiments on the Mitchell and Lapata (2010) dataset, we present results the APT model and the untyped model, using composition by *union* and composition by *intersection*, with and without distributional inference. We compare our models with the best performing untyped VSMs of Mitchell and Lapata (2010), and Blacoe and Lapata (2012), the best performing APT model of Weir et al. (2016), as well as with the recently published state-of-the-art methods by Hashimoto et al. (2014), and Wieting et al. (2015), who are using neural network based approaches. For our models, we use the static top n approach as neighbour retrieval function and tune the remaining parameters, the SPPMI shift k (1, 5, 10, 40, 100) and the number of neighbours (10, 30, 50, 100, 500, 1000, 5000), for both model types, and the sliding window size for the untyped VSM (1, 2, 5), on the development portion of the Mitchell and Lapata (2010) dataset. We keep the vector configuration (k and window size) fixed for all phrase types and only tune the number of neighbours used for DI individually. The best vector configuration for the APT model is achieved with $k = 10$ and for the untyped VSM with $k = 1$. For composition by *intersection* best performance on the dev set was achieved with 1000 neighbours for ANs, 10 for NNs and 50 for VOs with DI. For composition by *union*, top performance was obtained with 100 neighbours for ANs, 30 neighbours for NNs and 50 for VOs. The best results for the untyped model on the dev set are achieved with a symmetric window size of 1 and using 5000 neighbours for ANs, 10 for NNs and 1000 for VOs with composition by pointwise multiplication, and 30 neighbours for ANs, 5000 for NNs and 5000 for VOs for composition by pointwise addition. The validated numbers of neighbours on the development set show that the problem of missing information appears to be more severe for semantic composition than for word similarity tasks. Even though a neighbour at rank 1000 or lower does not appear to have a close relationship to the target word,

Model	Adjective-Noun	Noun-Noun	Verb-Object	Average
APT – *union*	0.45 (0.45)	0.45 (0.43)	0.38 (0.37)	0.43 (0.42)
APT – *intersect*	0.50 (0.38)	0.49 (0.44)	0.43 (0.36)	0.47 (0.39)
Untyped VSM – *addition*	0.46 (0.46)	0.40 (0.41)	0.38 (0.33)	0.41 (0.40)
Untyped VSM – *multiplication*	0.46 (0.42)	0.48 (0.45)	0.40 (0.39)	0.45 (0.42)
Mitchell and Lapata (2010) (untyped VSM & *multiplication*)	0.46	0.49	0.37	0.44
Blacoe and Lapata (2012) (untyped VSM & *multiplication*)	0.48	**0.50**	0.35	0.44
Hashimoto et al. (2014) (PAS-CLBLM & Add_{nl})	**0.52**	0.46	0.45	**0.48**
Wieting et al. (2015) (Paragram word embeddings & *RNN*)	0.51	0.40	**0.50**	0.47
Weir et al. (2016) (APT & *union*)	0.45	0.42	0.42	0.43

Table 7: Results for the Mitchell and Lapata (2010) dataset. Results in brackets denote the performance of the respective models without the use of distributional inference. Underlined means best within group, boldfaced means best overall.

it still can contribute useful co-occurrence information not observed in the original vector.

Table 7 shows that composition by *intersection* with distributional inference considerably improves upon the best results for APT models without distributional inference and for untyped count-based models, and is competitive with the state-of-the-art neural network based models of Hashimoto et al. (2014) and Wieting et al. (2015). Distributional inference also improves upon the performance of an untyped VSM where composition by pointwise multiplication is outperforming the models of Mitchell and Lapata (2010), and Blacoe and Lapata (2012). Table 7 furthermore shows that DI has a smaller effect on the APT model based on composition by *union* and the untyped model based on composition by pointwise addition. The reason, as pointed out in the discussion for Table 5, is that the composition function has no disambiguating effect and thus cannot eliminate unrelated neighbours introduced by distributional inference. An intersective composition function on the other hand is able to perform the disambiguation locally in any given phrasal context. This furthermore suggests that for the APT model it is not necessary to explicitly model different word senses in separate vectors, as composition by *intersection* is able to disambiguate any word in context individually. Unlike the models of Hashimoto et al. (2014) and Wieting et al. (2015), the elementary word representations, as well as the representations for composed phrases and the composition process in our models are fully interpretable[2].

6 Conclusion and Future Work

One of the major challenges in count-based models is dealing with extreme sparsity and missing information. This paper contributes a number of findings relating to this challenge, in particular a simple unsupervised algorithm for enriching sparse word representations by leveraging its distributional neighbourhood. We have demonstrated its benefit to typed and untyped vector space models on a range of word similarity datasets. We have shown that distributional inference improves the performance of typed and untyped VSMs for semantic composition and that our APT model is competitive with the state-of-the-art for adjective-noun, noun-noun and verb-object compositions while being fully interpretable. With our method, we are able to bridge the gap in performance between low-dimensional non-interpretable and high-dimensional interpretable representations. Lastly, we have investigated the different behaviour of composition by *union* and composition by *intersection* and have shown that an *intersective* composition function, together with distributional inference, has the ability to locally disambiguate the meaning of a phrase.

In future work we aim to scale our approach to semantic composition with distributional inference to longer phrases and full sentences. We furthermore plan to investigate whether the number of neighbours required for improving elementary vector representations remains as high for other compositional tasks and longer phrases as in this study.

[2]We release the APT vectors and our code at `https://github.com/tttthomasssss/apt-toolkit`.

Acknowledgments

This work was funded by UK EPSRC project EP/IO37458/1 *"A Unified Model of Compositional and Distributional Compositional Semantics: Theory and Applications"*. We would like to thank Miroslav Batchkarov for valuable discussions on earlier drafts of this paper and our anonymous reviewers for their helpful comments.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of NAACL-HLT*, pages 19–27, Boulder, Colorado, June. Association for Computational Linguistics.

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

Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *Proceedings of GEMS Workshop*, GEMS '11, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of EMNLP*, pages 1183–1193, Cambridge, MA, October. Association for Computational Linguistics.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of EMNLP*, pages 546–556, Jeju Island, Korea, July. Association for Computational Linguistics.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47, January.

John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, pages 510–526.

John A. Bullinaria and Joseph P. Levy. 2012. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd. *Behavior Research Methods*, 44(3):890–907.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, March.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *CoRR*, abs/1003.4394.

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

Ido Dagan, Fernando Pereira, and Lillian Lee. 1994. Similarity-based estimation of word cooccurrence probabilities. In *Proceedings of ACL*, pages 272–278, Las Cruces, New Mexico, USA, June. Association for Computational Linguistics.

Ido Dagan, Lillian Lee, and Fernando Pereira. 1997. Similarity-based methods for word sense disambiguation. In *Proceedings of ACL*, pages 56–63, Madrid, Spain, July. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of LREC*, pages 4585–4592, Reykjavik, Iceland, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L14-1045.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *J. Amer. Soc. Inf. Sci.*, 41(6):391–407.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of EMNLP*, pages 897–906, Honolulu, Hawaii, October. Association for Computational Linguistics.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of WWW*, WWW '01, pages 406–414, New York, NY, USA. ACM.

Gottlob Frege. 1884. *Die Grundlagen der Arithmetik: Eine logisch mathematische Untersuchung über den Begriff der Zahl*. W. Koebner.

Edward Grefenstette, Georgiana Dinu, Yao-Zhong Zhang, Mehrnoosh Sadrzadeh, and Marco Baroni. 2013. Multi-step regression learning for compositional distributional semantics. *Proceedings of IWCS*.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Norwell, MA, USA.

Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. 2014. Jointly learning word representations and composition functions using predicate-argument structures. In *Proceedings of*

EMNLP, pages 1544–1555, Doha, Qatar, October. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, December.

Gabriella Lapesa and Stefan Evert. 2014. A large scale evaluation of distributional semantic models: Parameters, interactions and model selection. *TACL*, 2:531–545.

Phong Le and Willem Zuidema. 2015. The forest convolutional network: Compositional distributional semantics with a neural chart and without binarization. In *Proceedings of EMNLP*, pages 1155–1164, Lisbon, Portugal, September. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proceedings of NIPS*, pages 2177–2185.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of ACL*, pages 768–774, Montreal, Quebec, Canada, August. Association for Computational Linguistics.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL - System Demonstrations*, pages 55–60.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *In Proceedings of ACL-08: HLT*, pages 236–244.

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

Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2015. Discriminative neural sentence modeling by tree-based convolution. In *Proceedings of EMNLP*, pages 2315–2325, Lisbon, Portugal, September. Association for Computational Linguistics.

Yoshiki Niwa and Yoshihiko Nitta. 1994. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of Coling*, COLING '94, pages 304–309, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Sebastian Padó, Jan Šnajder, and Britta Zeller. 2013. Derivational smoothing for syntactic distributional semantics. In *Proceedings of ACL*, pages 731–735, Sofia, Bulgaria, August. Association for Computational Linguistics.

Denis Paperno, Nghia The Pham, and Marco Baroni. 2014. A practical and linguistically-motivated approach to compositional distributional semantics. In *Proceedings of ACL*, pages 90–99, Baltimore, Maryland, June. Association for Computational Linguistics.

Emanuel Parzen. 1962. On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3):1065–1076, 09.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Magnus Sahlgren. 2006. *The Word-space model*. Ph.D. thesis, University of Stockholm (Sweden).

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 IJCNLP*, pages 489–497, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP*, pages 1201–1211, Jeju Island, Korea, July. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of ACL*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.

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

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2011. Word meaning in context: A simple and effective vector model. In *Proceedings of IJCNLP*, pages 1134–1143, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January.

Julie Weeds, David Weir, and Jeremy Reffin. 2014. Distributional composition using higher-order dependency vectors. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality*, pages 11–20, Gothenburg, Sweden, April. Association for Computational Linguistics.

David Weir, Julie Weeds, Jeremy Reffin, and Thomas Kober. 2016. Aligning packed dependency trees: a theory of composition for distributional semantics. *Computational Linguistics*, in press.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *TACL*, 3:345–358.

Benjamin Wilson. 2015. The unknown perils of mining wikipedia. https://blog.lateral.io/2015/06/the-unknown-perils-of-mining-wikipedia/, June.

Fabio Massimo Zanzotto, Lorenzo Ferrone, and Marco Baroni. 2015. Squibs: When the whole is not greater than the combination of its parts: A "decompositional" look at compositional distributional semantics. *Computational Linguistics*, 41(1):165–173.