Evaluating Neural Word Representations in Tensor-Based Compositional Settings

Dmitrijs Milajevs[1] **Dimitri Kartsaklis**[2] **Mehrnoosh Sadrzadeh**[1] **Matthew Purver**[1]

[1]Queen Mary University of London
School of Electronic Engineering
and Computer Science
Mile End Road, London, UK
{d.milajevs,m.sadrzadeh,m.purver}@qmul.ac.uk

[2]University of Oxford
Department of Computer Science
Parks Road, Oxford, UK
dimitri.kartsaklis@cs.ox.ac.uk

Abstract

We provide a comparative study between neural word representations and traditional vector spaces based on co-occurrence counts, in a number of compositional tasks. We use three different semantic spaces and implement seven tensor-based compositional models, which we then test (together with simpler additive and multiplicative approaches) in tasks involving verb disambiguation and sentence similarity. To check their scalability, we additionally evaluate the spaces using simple compositional methods on larger-scale tasks with less constrained language: paraphrase detection and dialogue act tagging. In the more constrained tasks, co-occurrence vectors are competitive, although choice of compositional method is important; on the larger-scale tasks, they are outperformed by neural word embeddings, which show robust, stable performance across the tasks.

1 Introduction

Neural word embeddings (Bengio et al., 2006; Collobert and Weston, 2008; Mikolov et al., 2013a) have received much attention in the distributional semantics community, and have shown state-of-the-art performance in many natural language processing tasks. While they have been compared with co-occurrence based models in simple similarity tasks at the word level (Levy et al., 2014; Baroni et al., 2014), we are aware of only one work that attempts a comparison of the two approaches in compositional settings (Blacoe and Lapata, 2012), and this is limited to additive and multiplicative composition, compared against composition via a neural autoencoder.

The purpose of this paper is to provide a more complete picture regarding the potential of neural word embeddings in compositional tasks, and meaningfully compare them with the traditional distributional approach based on co-occurrence counts. We are especially interested in investigating the performance of neural word vectors in compositional models involving general mathematical composition operators, rather than in the more task- or domain-specific deep-learning compositional settings they have generally been used with so far (for example, by Socher et al. (2012), Kalchbrenner and Blunsom (2013) and many others).

In particular, this is the first large-scale study to date that applies neural word representations in tensor-based compositional distributional models of meaning similar to those formalized by Coecke et al. (2010). We test a range of implementations based on this framework, together with additive and multiplicative approaches (Mitchell and Lapata, 2008), in a variety of different tasks. Specifically, we use the verb disambiguation task of Grefenstette and Sadrzadeh (2011a) and the transitive sentence similarity task of Kartsaklis and Sadrzadeh (2014) as small-scale focused experiments on pre-defined sentence structures. Additionally, we evaluate our vector spaces on paraphrase detection (using the Microsoft Research Paraphrase Corpus of Dolan et al. (2005)) and dialogue act tagging using the Switchboard Corpus (see e.g. (Stolcke et al., 2000)).

In all of the above tasks, we compare the neural word embeddings of Mikolov et al. (2013a) with two vector spaces both based on co-occurrence counts and produced by standard distributional techniques, as described in detail below. The general picture we get from the results is that in almost all cases the neural vectors are more effective than the traditional approaches.

We proceed as follows: Section 2 provides a concise introduction to distributional word representations in natural language processing. Section

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 708–719,
October 25-29, 2014, Doha, Qatar.

3 takes a closer look to the subject of compositionality in vector space models of meaning and describes the range of compositional operators examined here. In Section 4 we provide details about the vector spaces used in the experiments. Our experimental work is described in detail in Section 5, and the results are discussed in Section 6. Finally, Section 7 provides conclusions.

2 Meaning representation

There are several approaches to the representation of word, phrase and sentence meaning. As natural languages are highly creative and it is very rare to see the same sentence twice, any practical approach dealing with large text segments must be *compositional*, constructing the meaning of phrases and sentences from their constituent parts. The ideal method would therefore express not only the similarity in meaning between those constituent parts, but also between the results of their composition, and do this in ways which fit with linguistic structure and generalisations thereof.

Formal semantics Formal approaches to the semantics of natural language have long built upon the classical idea of compositionality – that the meaning of a sentence is a function of the meanings of its parts (Frege, 1892). In compositional type-logical approaches, predicate-argument structures representing phrases and sentences are built from their constituent parts by β-reduction within the lambda calculus framework (Montague, 1970): for example, given a representation of *John* as $john'$ and *sleeps* as $\lambda x.sleep'(x)$, the meaning of the sentence "John sleeps" can be constructed as $\lambda x.sleep'(x)(john') = sleep'(john')$. Given a suitable pairing between words and semantic representations of them, this method can produce structured sentential representations with broad coverage and good generalisability (see e.g. (Bos, 2008)). The above logical approach is extremely powerful because it can capture complex aspects of meaning such as quantifiers and their interaction (see e.g. (Copestake et al., 2005)), and enables inference using well studied and developed logical methods (see e.g. (Bos and Gabsdil, 2000)).

Distributional hypothesis However, such formal approaches are less able to express *similarity* in meaning. We would like to capture the intuition that while *John* and *Mary* are distinct, they are rather similar to each other (both of them are humans) and dissimilar to words such as *dog*, *pavement* or *idea*. The same applies at the phrase and sentence level: "dogs chase cats" is similar in meaning to "hounds pursue kittens", but less so to "cats chase dogs" (despite the lexical overlap).

Distributional methods provide a way to address this problem. By representing words and phrases as vectors or tensors in a (usually highly dimensional) vector space, one can express similarity in meaning via a suitable distance metric within that space (usually cosine distance); furthermore, composition can be modelled via suitable linear-algebraic operations.

Co-occurrence-based word representations One way to produce such vectorial representations is to directly exploit Harris (1954)'s intuition that semantically similar words tend to appear in similar contexts. We can construct a vector space in which the dimensions correspond to contexts, usually taken to be words as well. The word vector components can then be calculated from the frequency with which a word has co-occurred with the corresponding contexts in a window of words, with a predefined length.

Table 1 shows 5 3-dimensional vectors for the words *Mary*, *John*, *girl*, *boy* and *idea*. The words *philosophy*, *book* and *school* signify vector space dimensions. As the vector for *John* is closer to *Mary* than it is to *idea* in the vector space—a direct consequence of the fact that *John*'s contexts are similar to *Mary*'s and dissimilar to *idea*'s—we can infer that *John* is semantically more similar to *Mary* than to *idea*.

Many variants of this approach exist: performance on word similarity tasks has been shown to be improved by replacing raw counts with weighted values (e.g. mutual information)—see (Turney et al., 2010) and below for discussion, and (Kiela and Clark, 2014) for a detailed comparison.

	philosophy	book	school
Mary	0	10	22
John	4	60	59
girl	0	19	93
boy	0	12	164
idea	10	47	39

Table 1: Word co-occurrence frequencies extracted from the BNC (Leech et al., 1994).

Neural word embeddings Deep learning techniques exploit the distributional hypothesis differently. Instead of relying on observed co-occurrence frequencies, a neural language model is trained to maximise some objective function related to e.g. the probability of observing the surrounding words in some context (Mikolov et al., 2013b):

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \qquad (1)$$

Optimizing the above function, for example, produces vectors which maximise the conditional probability of observing words in a context around the target word w_t, where c is the size of the training window, and $w_1 w_2, \cdots w_T$ a sequence of words forming a training instance. Therefore, the resulting vectors will capture the distributional intuition and can express degrees of lexical similarity.

This method has an obvious advantage compared to co-occurrence method: since now the context is *predicted*, the model in principle can be much more robust in data sparsity problems, which is always an important issue for co-occurrence word spaces. Additionally, neural vectors have also proven successful in other tasks (Mikolov et al., 2013c), since they seem to encode not only attributional similarity (the degree to which similar words are close to each other), but also relational similarity (Turney, 2006). For example, it is possible to extract the singular:plural relation (*apple*:*apples*, *car*:*cars*) using vector subtraction:

$$\overrightarrow{apple} - \overrightarrow{apples} \approx \overrightarrow{car} - \overrightarrow{cars}$$

Perhaps even more importantly, semantic relationships are preserved in a very intuitive way:

$$\overrightarrow{king} - \overrightarrow{man} \approx \overrightarrow{queen} - \overrightarrow{woman}$$

allowing the formation of analogy queries similar to $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} = ?$, obtaining $\overrightarrow{queen}$ as the result.[1]

Both neural and co-occurrence-based approaches have advantages over classical formal approaches in their ability to capture lexical semantics and degrees of similarity; their success at extending this to the sentence level and to more complex semantic phenomena, though, depends on their applicability within compositional models, which is the subject of the next section.

3 Compositional models

Compositional distributional models represent meaning of a sequence of words by a vector, obtained by combining meaning vectors of the words within the sequence using some vector composition operation. In a general classification of these models, one can distinguish between three broad cases: simplistic models which combine word vectors irrespective of their order or relation to one another, models which exploit linear word order, and models which use grammatical structure.

The first approach combines word vectors by vector addition or point-wise multiplication (Mitchell and Lapata, 2008)—as this is independent of word order, it cannot capture the difference between the two sentences "dogs chase cats" and "cats chase dogs". The second approach has generally been implemented using some form of deep learning, and captures word order, but not by necessarily caring about the grammatical structure of the sentence. Here, one works by recursively building and combining vectors for subsequences of words within the sentence using e.g. autoencoders (Socher et al., 2012) or convolutional filters (Kalchbrenner et al., 2014). We do not consider this approach in this paper. This is because, as mentioned in the introduction, their vectors and composition operators are task-specific. These are trained directly to achieve specific objectives in certain pre-determined tasks. We are interested in vector and composition operators that work for *any* compositional task, and which can be combined with results in linguistics and formal semantics to provide generalisable models that can canonically extend to complex semantic phenomena. The third (i.e. the grammatical) approach promises a way to achieve this, and has been instantiated in various ways in the work of Baroni and Zamparelli (2010),Grefenstette and Sadrzadeh (2011a), and Kartsaklis et al. (2012).

General framework Formally, we can specify the vector representation of a word sequence $w_1 w_2 \cdots w_n$ as the vector $\overrightarrow{s} = \overrightarrow{w_1} \star \overrightarrow{w_2} \star \cdots \star \overrightarrow{w_n}$, where $\star$ is a vector operator, such as addition $+$, point-wise multiplication $\odot$, tensor product $\otimes$, or matrix multiplication $\times$.

[1] Levy et al. (2014) improved Mikolov et al. (2013c)'s method of retrieving relational similarities by changing the underlying objective function.

In the simplest compositional models (the first approach described above), $\star$ is $+$ or $\odot$, e.g. see (Mitchell and Lapata, 2008). Grammar-based compositional models (the third approach) are based on a generalisation of the notion of vectors, known as *tensors*. Whereas a vector $\overrightarrow{v}$ is an element of an atomic vector space V, a tensor $\overline{z}$ is an element of a tensor space $V \otimes W \otimes \cdots \otimes Z$. The number of tensored spaces is referred to by the *order* of the space. Using a general duality theorem from multi-linear algebra (Bourbaki, 1989), it follows that tensors are in one-one correspondence with multi-linear maps, that is we have:

$$\overline{z} \in V \otimes W \otimes \cdots \otimes Z \;\cong\; f_{\overline{z}} \colon V \to W \to \cdots \to Z$$

In such a tensor-based formalism, meanings of nouns are vectors and meanings of predicates such as adjectives and verbs are tensors. Meaning of a string of words is obtained by applying the compositions of multi-linear map duals of the tensors to the vectors. For the sake of demonstration, take the case of an intransitive sentence "Sbj Verb"; the meaning of the subject is a vector $\overrightarrow{\text{Sbj}} \in V$ and the meaning of the intransitive verb is a tensor $\overline{\text{Verb}} \in V \otimes W$. Meaning of the sentence is obtained by applying $f_{\overline{Verb}}$ to $\overrightarrow{\text{Sbj}}$, as follows:

$$\overrightarrow{\text{Sbj Verb}} = f_{\overline{Verb}}(\overrightarrow{\text{Sbj}})$$

By tensor-map duality, the above becomes equivalent to the following, where composition has now become the familiar notion of matrix multiplication, that is $\star$ is $\times$:

$$\overline{\text{Verb}} \times \overrightarrow{\text{Sbj}}$$

In general and for words with tensors of order higher than two, $\star$ becomes a generalisation of $\times$, referred to by *tensor contraction*, see e.g. Kartsaklis and Sadrzadeh (2013). Since the creation and manipulation of tensors of order higher than 2 is difficult, one can work with simplified versions of tensors, faithful to their underlying mathematical basis; these have found intuitive interpretations, e.g. see Grefenstette and Sadrzadeh (2011a), Kartsaklis and Sadrzadeh (2014). In such cases, $\star$ becomes a combination of a range of operations such as $\times$, $\otimes$, $\odot$, and $+$.

Specific models In the current paper we will experiment with a variety of models. In Table 2, we present these models in terms of their composition operators and a reference to the main paper in which each model was introduced. For the simple compositional models the sentence is a string of any number of words; for the grammar-based models, we consider simple transitive sentences "Sbj Verb Obj" and introduce the following abbreviations for the concrete method used to build a tensor for the verb:

1. $\overline{\text{Verb}}$ is a verb matrix computed using the formula $\sum_i \overrightarrow{\text{Sbj}_i} \otimes \overrightarrow{\text{Obj}_i}$, where $\overrightarrow{\text{Sbj}_i}$ and $\overrightarrow{\text{Obj}_i}$ are the subjects and objects of the verb across the corpus. These models are referred to by *relational* (Grefenstette and Sadrzadeh, 2011a); they are generalisations of predicate semantics of transitive verbs, from pairs of individuals to pairs of vectors. The models reduce the order 3 tensor of a transitive verb to an order 2 tensor (i.e. a matrix).

2. $\widetilde{\text{Verb}}$ is a verb matrix computed using the formula $\overrightarrow{\text{Verb}} \otimes \overrightarrow{\text{Verb}}$, where $\overrightarrow{\text{Verb}}$ is the distributional vector of the verb. These models are referred to by *Kronecker*, which is the term sometimes used to denote the outer product of tensors (Grefenstette and Sadrzadeh, 2011b). This models also reduces the order 3 tensor of a transitive verb to an order 2 tensor.

3. The models of the last five lines of the table use the so-called *Frobenius* operators from categorical compositional distributional semantics (Kartsaklis et al., 2012) to expand the relational matrices of verbs from order 2 to order 3. The expansion is obtained by either copying the dimension of the subject into the space provided by the third tensor, hence referred to by *Copy-Sbj*, or copying the dimension of the object in that space, hence referred to by *Copy-Obj*; furthermore, we can take addition, multiplication, or outer product of these, which are referred to by *Frobenius-Add*, *Frobenius-Mult*, and *Frobenius-Outer* (Kartsaklis and Sadrzadeh, 2014).

4 Semantic word spaces

Co-occurrence-based vector space instantiations have received a lot of attention from the scientific community (refer to (Kiela and Clark, 2014; Polajnar and Clark, 2014) for recent studies). We instantiate two co-occurrence-based vectors spaces with different underlying corpora and weighting schemes.

Method	Sentence	Linear algebraic formula	Reference
Addition	$w_1 w_2 \cdots w_n$	$\overrightarrow{w_1} + \overrightarrow{w_2} + \cdots + \overrightarrow{w_n}$	Mitchell and Lapata (2008)
Multiplication	$w_1 w_2 \cdots w_n$	$\overrightarrow{w_1} \odot \overrightarrow{w_2} \odot \cdots \odot \overrightarrow{w_n}$	Mitchell and Lapata (2008)
Relational	Sbj Verb Obj	$\overline{\text{Verb}} \odot (\overrightarrow{\text{Sbj}} \otimes \overrightarrow{\text{Obj}})$	Grefenstette and Sadrzadeh (2011a)
Kronecker	Sbj Verb Obj	$\widetilde{\text{Verb}} \odot (\overrightarrow{\text{Sbj}} \otimes \overrightarrow{\text{Obj}})$	Grefenstette and Sadrzadeh (2011b)
Copy object	Sbj Verb Obj	$\overrightarrow{\text{Sbj}} \odot (\overline{\text{Verb}} \times \overrightarrow{\text{Obj}})$	Kartsaklis et al. (2012)
Copy subject	Sbj Verb Obj	$\overrightarrow{\text{Obj}} \odot (\overline{\text{Verb}}^{\mathsf{T}} \times \overrightarrow{\text{Sbj}})$	Kartsaklis et al. (2012)
Frob. add.	Sbj Verb Obj	$(\overrightarrow{\text{Sbj}} \odot (\overline{\text{Verb}} \times \overrightarrow{\text{Obj}})) + (\overrightarrow{\text{Obj}} \odot (\overline{\text{Verb}}^{\mathsf{T}} \times \overrightarrow{\text{Sbj}}))$	Kartsaklis and Sadrzadeh (2014)
Frob. mult.	Sbj Verb Obj	$(\overrightarrow{\text{Sbj}} \odot (\overline{\text{Verb}} \times \overrightarrow{\text{Obj}})) \odot (\overrightarrow{\text{Obj}} \odot (\overline{\text{Verb}}^{\mathsf{T}} \times \overrightarrow{\text{Sbj}}))$	Kartsaklis and Sadrzadeh (2014)
Frob. outer	Sbj Verb Obj	$(\overrightarrow{\text{Sbj}} \odot (\overline{\text{Verb}} \times \overrightarrow{\text{Obj}})) \otimes (\overrightarrow{\text{Obj}} \odot (\overline{\text{Verb}}^{\mathsf{T}} \times \overrightarrow{\text{Sbj}}))$	Kartsaklis and Sadrzadeh (2014)

Table 2: Compositional methods.

GS11 Our first word space is based on a typical configuration that has been used in the past extensively for compositional distributional models (see below for details), so it will serve as a useful baseline for the current work. In this vector space, the co-occurrence counts are extracted from the British National Corpus (BNC) (Leech et al., 1994). As basis words, we use the most frequent nouns, verbs, adjectives and adverbs (POS tags `SUBST`, `VERB`, `ADJ` and `ADV` in the BNC XML distribution[2]). The vector space is lemmatized, that is, it contains only "canonical" forms of words.

In order to weight the raw co-occurrence counts, we use positive point-wise mutual information (PPMI). The component value for a target word t and a context word c is given by:

$$\text{PPMI}(t, c) = \max\left(0, \log \frac{p(c|t)}{p(c)}\right)$$

where $p(c|t)$ is the probability of word c given t in a symmetric window of length 5 and $p(c)$ is the probability of c overall.

Vector spaces based on point-wise mutual information (or variants thereof) have been successfully applied in various distributional and compositional tasks; see e.g. Grefenstette and Sadrzadeh (2011a), Mitchell and Lapata (2008), Levy et al. (2014) for details. PPMI has been shown to achieve state-of-the-art results (Levy et al., 2014) and is suggested by the review of Kiela and Clark (2014). Our use here of the BNC as a corpus and the window length of 5 is based on previous use and better performance of these parameters in a number of compositional experiments (Grefenstette and Sadrzadeh, 2011a; Grefenstette and Sadrzadeh, 2011b; Mitchell and Lapata, 2008; Kartsaklis et al., 2012).

KS14 In this variation, we train a vector space from the ukWaC corpus[3] (Ferraresi et al., 2008), originally using as a basis the 2,000 content words with the highest frequency (but excluding a list of stop words as well as the 50 most frequent content words since they exhibit low information content). The vector space is again lemmatized. As context we consider a 5-word window from either side of the target word, while as our weighting scheme we use local mutual information (i.e. point-wise mutual information multiplied by raw counts). In a further step, the vector space was normalized and projected onto a 300-dimensional space using singular value decomposition (SVD).

In general, dimensionality reduction produces more compact word representations that are robust against potential noise in the corpus (Landauer and Dumais, 1997; Schütze, 1997). SVD has been shown to perform well on a variety of tasks similar to ours (Baroni and Zamparelli, 2010; Kartsaklis and Sadrzadeh, 2014).

Neural word embeddings (NWE) For our neural setting, we used the skip-gram model of Mikolov et al. (2013b) trained with negative sampling. The specific implementation that was tested in our experiments was a 300-dimensional vector space learned from the Google News corpus and provided by the `word2vec`[4] toolkit. Furthermore, the `gensim` library (Řehůřek and Sojka, 2010) was used for accessing the vectors. On the contrary with the previously described co-

[2] `http://www.natcorp.ox.ac.uk/`

[3] `http://wacky.sslmit.unibo.it/`

[4] `https://code.google.com/p/word2vec/`

occurrence vector spaces, this version is *not* lemmatized.

The negative sampling method improves the objective function of Equation 1 by introducing negative examples to the training algorithm. Assume that the probability of a specific (c, t) pair of words (where t is a target word and c another word in the same context with t), coming from the training data, is denoted as $p(D = 1|c, t)$. The objective function is then expressed as follows:

$$\prod_{(c,t)\in D} p(D = 1|c, t) \quad (2)$$

That is, the goal is to set the model parameters in a way that maximizes the probability of all observations coming from the training data. Assume now that D' is a set of randomly selected incorrect (c', t') pairs that do not occur in D, then Equation 2 above can be recasted in the following way:

$$\prod_{(c,t)\in D} p(D = 1|c, t) \prod_{(c',t')\in D'} p(D = 0|c', t') \quad (3)$$

In other words, the model tries to distinguish a target word t from random draws that come from a noise distribution. In the implementation we used for our experiments, c is always selected from a 5-word window around t. More details about the negative sampling approach can be found in (Mikolov et al., 2013b); the note of Goldberg and Levy (2014) also provides an intuitive explanation of the underlying setting.

5 Experiments

Our experiments explore the use of the vector spaces above, together with the compositional operators described in Section 3, in a range of tasks all of which require semantic composition: verb sense disambiguation; sentence similarity; paraphrasing; and dialogue act tagging.

5.1 Disambiguation

We use the transitive verb disambiguation dataset described in Grefenstette and Sadrzadeh (2011a)[5]. This dataset consists of ambiguous transitive verbs together with their arguments, landmark verbs that identify one of the verb senses, and human judgements that specify how similar is the disambiguated sense of the verb in the given context to one of the landmarks. This is similar to the intransitive dataset described in (Mitchell and Lapata, 2008). Consider the sentence "system meets specification"; here, *meets* is the ambiguous transitive verb, and *system* and *specification* are its arguments in this context. Possible landmarks for *meet* are *satisfy* and *visit*; for this sentence, the human judgements show that the disambiguated meaning of the verb is more similar to the landmark *satisfy* and less similar to *visit*.

The task is to estimate the similarity of the sense of a verb in a context with a given landmark. To get our similarity measures, we compose the verb with its arguments using one of our compositional models; we do the same for the landmark and then compute the cosine similarity of the two vectors. We evaluate the performance by averaging the human judgements for the same verb, argument and landmark entries, and calculating the Spearman's correlation between the average values and the cosine scores. As a baseline, we compare this with the correlation produced by using only the verb vector, without composing it with its arguments.

Table 3 shows the results of the experiment. NWE *copy-object* composition yields the best correlation with the human judgements, and top performance across all vector spaces and models with a Spearman ρ of 0.456. For the KS14 space, the best result comes from *Frobenius outer* (0.350),

Method	GS11	KS14	NWE
Verb only	0.212	0.325	0.107
Addition	0.103	0.275	0.149
Multiplication	0.348	0.041	0.095
Kronecker	0.304	0.176	0.117
Relational	0.285	0.341	0.362
Copy subject	0.089	0.317	0.131
Copy object	0.334	0.331	**0.456**
Frobenius add.	0.261	0.344	0.359
Frobenius mult.	0.233	0.341	0.239
Frobenius outer	0.284	0.350	0.375

Table 3: Spearman ρ correlations of models with human judgements for the word sense disambiguation task. The best result (NWE Copy object) outperforms the nearest co-occurrence-based competitor (KS14 Frobenius outer) with a statistically significant difference ($p < 0.05$, t-test).

[5]This and the sentence similarity dataset are available at `http://www.cs.ox.ac.uk/activities/compdistmeaning/`

while the best operator for the GS11 space is *point-wise multiplication* (0.348).

For simple point-wise composition, only multiplicative GS11 and additive NWE improve over their corresponding verb-only baselines (but both perform worse than the KS14 baseline). With tensor-based composition in co-occurrence based spaces, *copy subject* yields lower results than the corresponding baselines. Other composition methods, except *Kronecker* for KS14, improve over the verb-only baselines. Finally we should note that, despite the small training corpus, the GS11 vector space performs comparatively well: for instance, *Kronecker* model improves the previously reported score of 0.28 (Grefenstette and Sadrzadeh, 2011b).

5.2 Sentence similarity

In this experiment we use the transitive sentence similarity dataset described in Kartsaklis and Sadrzadeh (2014). The dataset consists of transitive sentence pairs and a human similarity judgement[6]. The task is to estimate a similarity measure between two sentences. As in the disambiguation task, we first compose word vectors to obtain sentence vectors, then compute cosine similarity of them. We average the human judgements for identical sentence pairs to compute a correlation with cosine scores.

Table 4 shows the results. Again, the best performing vector space is KS14, but this time with *addition*: the Spearman ρ correlation score with averaged human judgements is 0.732. Addition was the means for the other vector spaces to achieve top performance as well: GS11 and NWE got 0.682 and 0.689 respectively.

None of the models in tensor-based composition outperformed addition. KS14 performs worse with tensor-based methods here than in the other vector spaces. However, GS11 and NWE, except *copy subject* for both of them and *Frobenius multiplication* for NWE, improved over their verb-only baselines.

5.3 Paraphrasing

In this experiment we evaluate our vector spaces on a mainstream paraphrase detection task.

[6]The textual content of this dataset is the same as that of (Kartsaklis and Sadrzadeh, 2013), the difference is that the dataset of (Kartsaklis and Sadrzadeh, 2014) has updated human judgements whereas the previous dataset used the original annotations of the intransitive dataset of (Mitchell and Lapata, 2010).

Method	**GS11**	**KS14**	**NWE**
Verb only	0.491	0.602	0.561
Addition	0.682	**0.732**	0.689
Multiplication	0.597	0.321	0.341
Kronecker	0.581	0.408	0.561
Relational	0.558	0.437	0.618
Copy subject	0.370	0.448	0.405
Copy object	0.571	0.306	0.655
Frobenius add.	0.566	0.460	0.585
Frobenius mult.	0.525	0.226	0.387
Frobenius outer	0.560	0.439	0.622

Table 4: Results for sentence similarity. There is no statistically significant difference between KS14 addition and NWE addition (the second best result).

Specifically, we get classification results on the Microsoft Research Paraphrase Corpus paraphrase corpus (Dolan et al., 2005) working in the following way: we construct vectors for the sentences of each pair; if the cosine similarity between the two sentence vectors exceeds a certain threshold, the pair is classified as a paraphrase, otherwise as not a paraphrase. For this experiment and that of Section 5.4 below, we investigate only the addition and point-wise multiplication compositional models, since at their current stage of development tensor-based models can only efficiently handle sentences of fixed structure. Nevertheless, the simple point-wise compositional models still allow for a direct comparison of the vector spaces, which is the main goal of this paper.

For each vector space and model, a number of different thresholds were tested on the first 2000 pairs of the training set, which we used as a development set; in each case, the best-performed threshold was selected for a *single* run of our "classifier" on the test set (1726 pairs). Additionally, we evaluate the NWE model with a lemmatized version of the corpus, so that the experimental setup is maximally similar for all vector spaces. The results are shown in the first part of Table 5.

Additive NWE gives the highest performance, with both lemmatized and un-lemmatized versions outperforming the GS11 and KS14 spaces. In the un-lemmatized case, the accuracy of our simple "classifier" (0.73) is close to state-of-the-art range. The state-of-the art result (0.77 accuracy

	Baseline		Co-occurrence				Neural word embeddings			
			GS11		KS14		Unlemmatized		Lemmatized	
Model	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
MSR addition	0.65	0.75	0.62	0.79	0.70	0.80	**0.73**	**0.82**	0.72	0.81
MSR multiplication			0.52	0.58	**0.66**	**0.80**	0.42	0.34	0.41	0.36
SWDA addition	0.60	0.58	0.35	0.35	0.40	0.35	**0.63**	**0.60**	0.44	0.40
SWDA multiplication			0.32	0.16	0.39	0.33	**0.58**	**0.53**	0.43	0.38

Table 5: Results for paraphrase detection (MSR) and dialog act tagging (SWDA) tasks. All top results significantly outperform corresponding nearest competitors (for accuracy): $p < 0.05$, χ^2 test.

and 0.84 F-score[7]) by the time of this writing has been obtained using 8 machine translation metrics and three constituent classifiers (Madnani et al., 2012).

The multiplicative model gives lower results than the additive model across all vector spaces. The KS14 vector space shows the steadiest performance, with a drop in accuracy of only 0.04 and no drop in F-score, while for the GS11 and NWE spaces both accuracy and F-score experienced drops by more than 0.20.

5.4 Dialogue act tagging

As our last experiment, we evaluate the word spaces on a dialogue act tagging task (Stolcke et al., 2000) over the Switchboard corpus (Godfrey et al., 1992). Switchboard is a collection of approximately 2500 dialogs over a telephone line by 500 speakers from the U.S. on predefined topics.[8]

The experiment pipeline follows (Milajevs and Purver, 2014). The input utterances are preprocessed so that the parts of interrupted utterances are concatenated (Webb et al., 2005). Disfluency markers and commas are removed from the utterance raw texts. For GS11 and KS14 the utterance tokens are POS-tagged and lemmatized; for NWE, we test the vectors in both a lemmatized and an un-lemmatized version of the corpus.[9] We split the training and testing utterances as suggested by Stolcke et al. (2000). Utterance vectors are then obtained as in the previous experiments; they are reduced to 50 dimensions using SVD and a k-nearest-neighbour classifier is trained on these reduced utterance vectors (the 5 closest neighbours by Euclidean distance are retrieved to make a classification decision). The results are shown in the second part of Table 5.

Un-lemmatized NWE *addition* gave the best accuracy (0.63) and F-score (0.60) (averaged over tag classes), i.e. similar results to (Milajevs and Purver, 2014)—although note that the dimensionality of our NWE vectors is 10 times lower than theirs. *Multiplicative* NWE outperformed the corresponding model in (Milajevs and Purver, 2014). In general, addition consistently outperforms multiplication for all the models. Lemmatization dramatically lowers tagging accuracy: the lemmatized GS11, KS14 and NWE models perform much worse than un-lemmatized NWE, suggesting that morphological features are important for this task.

6 Discussion

Previous comparisons of co-occurrence-based and neural word vector representations vary widely in their conclusions. While Baroni et al. (2014) conclude that "context-predicting models obtain a thorough and resounding victory against their count-based counterparts", this seems to contradict, at least at the first consideration, the more conservative conclusion of Levy et al. (2014) that "analogy recovery is not restricted to neural word embeddings [...] a similar amount of relational similarities can be recovered from traditional distributional word representations" and the findings of Blacoe and Lapata (2012) that "shallow approaches are as good as more computationally intensive alternatives" on phrase similarity and paraphrase detection tasks.

It seems clear that neural word embeddings have an advantage when used in tasks for which they have been trained; our main questions here are whether they outperform co-occurrence based alternatives across the board; and which approach lends itself better to composition using general mathematical operators. To partially an-

[7] F-scores use the standard definition $F = 2(precision * recall)/(precision + recall)$.

[8] The dataset and a Python interface to it are available at `http://compprag.christopherpotts.net/swda.html`

[9] We use `WordNetLemmatizer` of the NLTK library (Bird, 2006).

swer this question, we can compare model behaviour against the baselines in *isolation*.

For the disambiguation and sentence similarity tasks the baseline is the similarity between verbs only, ignoring the context—see above. For the paraphrase task, we take the global vector-based similarity reported in (Mihalcea et al., 2006): 0.65 accuracy and 0.75 F-score. For the dialogue act tagging task the baseline is the accuracy of the bag-of-unigrams model in (Milajevs and Purver, 2014): 0.60.

Sections 5.1 and 5.2 show that although the best choice of vector representation might vary, for small-scale tasks all methods give fairly competitive results. The choice of compositional operator seems to be more important and more task-specific: while a tensor-based operation (Frobenius copy-object) performs best for verb disambiguation, the best result for sentence similarity is achieved by a simple additive model, with all other compositional methods behaving worse than the verb-only baseline in the KS14 case. GS11 and NWE, on the other hand, outperform their baselines with a number of compositional methods, although both of them achieve lower performance than KS14 overall.

Based on only small-scale experiment results, one could conclude that there is little significant difference between the two ways of obtaining vectors. GS11 and NWE show similar behaviour in comparison to their baselines, while it is possible to tune a co-occurrence based vector space (KS14) and obtain the best result. Large scale tasks reveal another pattern: the GS11 vector space, which behaves stably on the small scale, drags behind the KS14 and NWE spaces in the paraphrase detection task. In addition, NWE consistently yields best results. Finally, only the NWE space was able to provide adequate results on the dialogue act tagging task. Table 6 summarizes model performance with regard to baselines.

Task	**GS11**	**KS14**	**NWE**
Disambiguation	+	+	**+**
Sentence similarity	+	**−**	+
Paraphrase	−	+	**+**
Dialog act tagging	−	−	**+**

Table 6: Summary of vector space performance against baselines. General improvement (cases where more than a half of the models perform better) and decrease with regard to a corresponding baseline is respectively marked by + and −. A bold value means that the model gave the best result in the task.

7 Conclusion

In this work we compared the performance of two co-occurrence-based semantic spaces with vectors learned by a neural network in compositional settings. We carried out two small-scale tasks (word sense disambiguation and sentence similarity) and two large-scale tasks (paraphrase detection and dialogue act tagging).

On small-scale tasks, where the sentence structures are predefined and relatively constrained, NWE gives better or similar results to count-based vectors. Tensor-based composition does not always outperform simple compositional operators, but for most of the cases gives results within the same range.

On large-scale tasks, neural vectors are more successful than the co-occurrence based alternatives. However, this study does not reveal whether this is because of their neural nature, or just because they are trained on a larger amount of data.

The question of whether neural vectors outperform co-occurrence vectors therefore requires further detailed comparison to be entirely resolved; our experiments suggest that this is indeed the case in large-scale tasks, but the difference in size and nature of the original corpora may be a confounding factor. In any case, it is clear that the neural vectors of `word2vec` package perform steadily off-the-shelf across a large variety of tasks. The size of the vector space (3 million words) and the available code-base that simplifies the access to the vectors, makes this set a good and safe choice for experiments in the future. Of course, even better performances can be achieved by training neural language models specifically for a given task (see e.g. Kalchbrenner et al. (2014)).

The choice of compositional operator (tensor-based or a simple point-wise operation) depends strongly on the task and dataset: tensor-based composition performed best with the verb disambiguation task, where the verb senses depend strongly on the arguments of the verb. However, it seems to depend less on the nature of the vectors itself: in the disambiguation task, tensor-based

composition proved best for both co-occurrence-based and neural vectors; in the sentence similarity task, where point-wise operators proved best, this was again true across vector spaces.

Acknowledgements

We would like to thank the three anonymous reviewers for their fruitful comments. Support by EPSRC grant EP/F042728/1 is gratefully acknowledged by Milajevs, Kartsaklis and Sadrzadeh. Purver is partly supported by ConCreTe: the project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

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*, pages 1183–1193. Association for Computational Linguistics.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Johan Bos and Malte Gabsdil. 2000. First-order inference and the interpretation of questions and answers. *Proceedings of Gotelog*, pages 43–50.

Johan Bos. 2008. Wide-coverage semantic analysis with boxer. In Johan Bos and Rodolfo Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, Research in Computational Semantics, pages 277–286. College Publications.

N. Bourbaki. 1989. *Commutative Algebra: Chapters 1-7*. Srpinger Verlag, Berlin/New York.

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

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332.

Bill Dolan, Chris Brockett, and Chris Quirk. 2005. Microsoft research paraphrase corpus. *Retrieved May*, 29:2013.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54.

Gottlob Frege. 1892. On sense and reference. *Ludlow (1997)*, pages 563–584.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 517–520. IEEE.

Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011a. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404. Association for Computational Linguistics.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011b. Experimenting with transitive verbs in a DisCoCat. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 62–66, Edinburgh, UK, July. Association for Computational Linguistics.

Z.S. Harris. 1954. Distributional structure. *Word*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126, Sofia, Bulgaria, August. Association for Computational Linguistics.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, June.

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. 2013. Prior disambiguation of word tensors for constructing sentence vectors. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNL)*, pages 1590–1601, Seattle, USA, October. Association for Computational Linguistics.

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. 2014. A study of entanglement in a categorical framework of natural language. In *Proceedings of the 11th Workshop on Quantum Physics and Logic (QPL)*, Kyoto, Japan, June.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. 2012. A unified sentence space for categorical distributional-compositional semantics: Theory and experiments. In *Proceedings of COLING 2012: Posters*, pages 549–558, Mumbai, India, December. The COLING 2012 Organizing Committee.

Douwe Kiela and Stephen Clark. 2014. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden, April. Association for Computational Linguistics.

T. Landauer and S. Dumais. 1997. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquision, Induction, and Representation of Knowledge. *Psychological Review*.

Geoffrey Leech, Roger Garside, and Michael Bryant. 1994. Claws4: the tagging of the british national corpus. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 622–628. Association for Computational Linguistics.

Omer Levy, Yoav Goldberg, and Israel Ramat-Gan. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning, Baltimore, Maryland, USA, June. Association for Computational Linguistics*.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190. Association for Computational Linguistics.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, volume 6, pages 775–780.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, pages 746–751.

Dmitrijs Milajevs and Matthew Purver. 2014. Investigating the contribution of distributional semantic information for dialogue act classification. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 40–47, Gothenburg, Sweden, April. Association for Computational Linguistics.

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

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1439.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Tamara Polajnar and Stephen Clark. 2014. Improving distributional semantic vectors through context selection and normalisation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 230–238, Gothenburg, Sweden, April. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. `http://is.muni.cz/publication/884893/en`.

Hinrich Schütze. 1997. Ambiguity resolution in natural language learning. csli. *Stanford, CA*, 4:12–36.

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.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Carol Van Ess-Dykema, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Peter D Turney, Patrick Pantel, et al. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

Peter D Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Nick Webb, Mark Hepple, and Yorick Wilks. 2005. Dialogue act classification based on intra-utterance features. In *Proceedings of the AAAI Workshop on Spoken Language Understanding*. Citeseer.