A Tensor-based Factorization Model of Semantic Compositionality

Tim Van de Cruys
IRIT – UMR 5505
CNRS
Toulouse, France
tim.vandecruys@irit.fr

Thierry Poibeau[⋆]
LaTTiCe – UMR 8094
CNRS & ENS
Paris, France
thierry.poibeau@ens.fr

Anna Korhonen
Computer Laboratory & DTAL[⋆]
University of Cambridge
United Kingdom
anna.korhonen@cl.cam.ac.uk

Abstract

In this paper, we present a novel method for the computation of compositionality within a distributional framework. The key idea is that compositionality is modeled as a multi-way interaction between latent factors, which are automatically constructed from corpus data. We use our method to model the composition of *subject verb object* triples. The method consists of two steps. First, we compute a latent factor model for nouns from standard co-occurrence data. Next, the latent factors are used to induce a latent model of three-way *subject verb object* interactions. Our model has been evaluated on a similarity task for transitive phrases, in which it exceeds the state of the art.

1 Introduction

In the course of the last two decades, significant progress has been made with regard to the automatic extraction of lexical semantic knowledge from large-scale text corpora. Most work relies on the distributional hypothesis of meaning (Harris, 1954), which states that words that appear within the same contexts tend to be semantically similar. A large number of researchers have taken this dictum to heart, giving rise to a plethora of algorithms that try to capture the semantics of words by looking at their distribution in text. Up till now, however, most work on the automatic acquisition of semantics only deals with individual words. The modeling of meaning beyond the level of individual words – i.e. the combination of words into larger units – is to a large degree left unexplored.

The principle of compositionality, often attributed to Frege, is the principle that states that the meaning of a complex expression is a function of the meaning of its parts and the way those parts are (syntactically) combined (Frege, 1892). It is the fundamental principle that allows language users to understand the meaning of sentences they have never heard before, by constructing the meaning of the complex expression from the meanings of the individual words. Recently, a number of researchers have tried to reconcile the framework of distributional semantics with the principle of compositionality (Mitchell and Lapata, 2008; Baroni and Zamparelli, 2010; Coecke et al., 2010; Socher et al., 2012). However, the absolute gains of the systems remain a bit unclear, and a simple method of composition – vector multiplication – often seems to produce the best results (Blacoe and Lapata, 2012).

In this paper, we present a novel method for the joint composition of a verb with its subject and direct object. The key idea is that compositionality is modeled as a multi-way interaction between latent factors, which are automatically constructed from corpus data. In order to adequately model the multi-way interaction between a verb and its subject and objects, a significant part of our method relies on tensor algebra. Additionally, our method makes use of a factorization model appropriate for tensors.

The remainder of the paper is structured as follows. In section 2, we give an overview of previous work that is relevant to the task of computing compositionality within a distributional framework. Section 3 presents a detailed description of our method, including an overview of the necessary mathematical

machinery. Section 4 illustrates our method with a number of detailed examples. Section 5 presents a quantitative evaluation, and compares our method to other models of distributional compositionality. Section 6, then, concludes and lays out a number of directions for future work.

2 Previous Work

In recent years, a number of methods have been developed that try to capture compositional phenomena within a distributional framework. One of the first approaches to tackle compositional phenomena in a systematic way is Mitchell and Lapata's (2008) approach. They explore a number of different models for vector composition, of which vector addition (the sum of each feature) and vector multiplication (the elementwise multiplication of each feature) are the most important. They evaluate their models on a noun-verb phrase similarity task, and find that the multiplicative model yields the best results, along with a weighted combination of the additive and multiplicative model.

Baroni and Zamparelli (2010) present a method for the composition of adjectives and nouns. In their model, an adjective is a linear function of one vector (the noun vector) to another vector (the vector for the adjective-noun pair). The linear transformation for a particular adjective is represented by a matrix, and is learned automatically from a corpus, using partial least-squares regression.

Coecke et al. (2010) present an abstract theoretical framework in which a sentence vector is a function of the Kronecker product of its word vectors, which allows for greater interaction between the different word features. A number of instantiations of the framework are tested experimentally in Grefenstette and Sadrzadeh (2011a) and Grefenstette and Sadrzadeh (2011b). The key idea is that relational words (e.g. adjectives or verbs) have a rich (multi-dimensional) structure that acts as a filter on their arguments. Our model uses an intuition similar to theirs.

Socher et al. (2012) present a model for compositionality based on recursive neural networks. Each node in a parse tree is assigned both a vector and a matrix; the vector captures the actual meaning of the constituent, while the matrix models the way it changes the meaning of neighbouring words and phrases.

Closely related to the work on compositionality is research on the computation of word meaning in context. Erk and Padó (2008, 2009) make use of selectional preferences to express the meaning of a word in context; the meaning of a word in the presence of an argument is computed by multiplying the word's vector with a vector that captures the inverse selectional preferences of the argument. Thater et al. (2009, 2010) extend the approach based on selectional preferences by incorporating second-order co-occurrences in their model. And Dinu and Lapata (2010) propose a probabilistic framework that models the meaning of words as a probability distribution over latent factors. This allows them to model contextualized meaning as a change in the original sense distribution. Dinu and Lapata use non-negative matrix factorization (NMF) to induce latent factors. Similar to their work, our model uses NMF – albeit in a slightly different configuration – as a first step towards our final factorization model.

In general, latent models have proven to be useful for the modeling of word meaning. One of the best known latent models of semantics is Latent Semantic Analysis (Landauer and Dumais, 1997), which uses singular value decomposition in order to automatically induce latent factors from term-document matrices. Another well known latent model of meaning, which takes a generative approach, is Latent Dirichlet Allocation (Blei et al., 2003).

Tensor factorization has been used before for the modeling of natural language. Giesbrecht (2010) describes a tensor factorization model for the construction of a distributional model that is sensitive to word order. And Van de Cruys (2010) uses a tensor factorization model in order to construct a three-way selectional preference model of verbs, subjects, and objects. Our underlying tensor factorization – Tucker decomposition – is the same as Giesbrecht's; and similar to Van de Cruys (2010), we construct a latent model of verb, subject, and object interactions. The way our model is constructed, however, is significantly different. The former research does not use any syntactic information for the construction of the tensor, while the latter makes use of a more restricted tensor factorization model, viz. parallel factor analysis (Harshman and Lundy, 1994).

The idea of modeling compositionality by means of tensor (Kronecker) product has been proposed in the literature before (Clark and Pulman, 2007; Coecke et al., 2010). However, the method presented here is the first that tries to capture compositional phenomena by exploiting the multi-way interactions between latent factors, induced by a suitable tensor factorization model.

3 Methodology

3.1 Mathematical preliminaries

The methodology presented in this paper requires a number of concepts and mathematical operations from tensor algebra, which are briefly reviewed in this section. The interested reader is referred to Kolda and Bader (2009) for a more thorough introduction to tensor algebra (including an overview of various factorization methods).

A *tensor* is a multidimensional array; it is the generalization of a matrix to more than two dimensions, or *modes*. Whereas matrices are only able to capture two-way co-occurrences, tensors are able to capture *multi-way* co-occurrences.[1] Following prevailing convention, tensors are represented by boldface Euler script notation ($\mathcal{X}$), matrices by boldface capital letters ($\mathbf{X}$), vectors by boldface lower case letters ($\mathbf{x}$), and scalars by italic letters (x).

The *n-mode product* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n \mathbf{U}$, and is defined elementwise as

$$(\mathcal{X} \times_n \mathbf{U})_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n} \quad (1)$$

The *Kronecker product* of matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$. The result is a matrix of size $(IK) \times (JL)$, and is defined by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \ldots & a_{IJ}\mathbf{B} \end{bmatrix} \quad (2)$$

[1] In this research, we limit ourselves to three-way co-occurrences of verbs, subject, and objects, modelled using a three-mode tensor.

A special case of the Kronecker product is the *outer product* of two vectors $\mathbf{a} \in \mathbb{R}^I$ and $\mathbf{b} \in \mathbb{R}^J$, denoted $\mathbf{a} \circ \mathbf{b}$. The result is a matrix $\mathbf{A} \in \mathbb{R}^{I \times J}$ obtained by multiplying each element of $\mathbf{a}$ with each element of $\mathbf{b}$.

Finally, the *Hadamard product*, denoted $\mathbf{A} * \mathbf{B}$, is the elementwise multiplication of two matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{I \times J}$, which produces a matrix that is equally of size $I \times J$.

3.2 The construction of latent noun factors

The first step of our method consists in the construction of a latent factor model for nouns, based on their context words. For this purpose, we make use of non-negative matrix factorization (Lee and Seung, 2000). Non-negative matrix factorization (NMF) minimizes an objective function – in our case the Kullback-Leibler (KL) divergence – between an original matrix $\mathbf{V}_{I \times J}$ and $\mathbf{W}_{I \times K}\mathbf{H}_{K \times J}$ (the matrix multiplication of matrices $\mathbf{W}$ and $\mathbf{H}$) subject to the constraint that all values in the three matrices be non-negative. Parameter K is set $\ll I, J$ so that a reduction is obtained over the original data. The factorization model is represented graphically in figure 1.

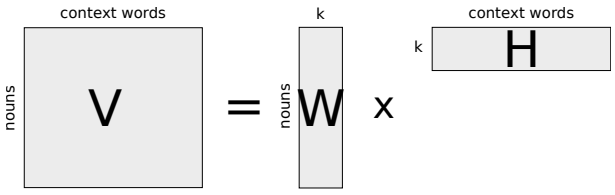

Figure 1: Graphical representation of NMF

NMF can be computed fairly straightforwardly, alternating between the two iterative update rules represented in equations 3 and 4. The update rules are guaranteed to converge to a local minimum in the KL divergence.

$$\mathbf{H}_{a\mu} \leftarrow \mathbf{H}_{a\mu} \frac{\sum_i \mathbf{W}_{ia} \frac{\mathbf{V}_{i\mu}}{(\mathbf{WH})_{i\mu}}}{\sum_k \mathbf{W}_{ka}} \quad (3)$$

$$\mathbf{W}_{ia} \leftarrow \mathbf{W}_{ia} \frac{\sum_\mu \mathbf{H}_{a\mu} \frac{\mathbf{V}_{i\mu}}{(\mathbf{WH})_{i\mu}}}{\sum_v \mathbf{H}_{av}} \quad (4)$$

3.3 Modeling multi-way interactions

In our second step, we construct a multi-way interaction model for *subject verb object* (*svo*) triples, based

on the latent factors induced in the first step. Our latent interaction model is inspired by a tensor factorization model called Tucker decomposition (Tucker, 1966), although our own model instantiation differs significantly. In order to explain our method, we first revisit Tucker decomposition, and subsequently explain how our model is constructed.

3.3.1 Tucker decomposition

Tucker decomposition is a multilinear generalization of the well-known singular value decomposition, used in Latent Semantic Analysis. It is also known as higher order singular value decomposition (HOSVD, De Lathauwer et al. (2000)). In Tucker decomposition, a tensor is decomposed into a core tensor, multiplied by a matrix along each mode. For a three-mode tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times L}$, the model is defined as

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \quad (5)$$

$$= \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r \quad (6)$$

Setting $P, Q, R \ll I, J, L$, the core tensor $\mathcal{G}$ represents a compressed, latent version of the original tensor $\mathcal{X}$; matrices $\mathbf{A} \in \mathbb{R}^{I \times P}$, $\mathbf{B} \in \mathbb{R}^{J \times Q}$, and $\mathbf{C} \in \mathbb{R}^{L \times R}$ represent the latent factors for each mode, while $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ indicates the level of interaction between the different latent factors. Figure 2 shows a graphical representation of Tucker decomposition.[2]

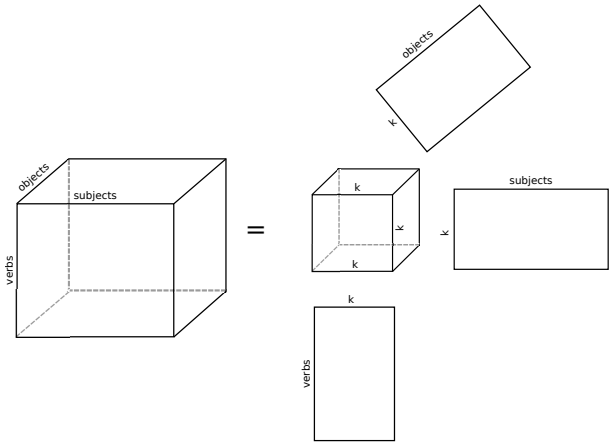

Figure 2: A graphical representation of Tucker decomposition

[2] where $P = Q = R = K$, i.e. the same number of latent factors K is used for each mode

3.3.2 Reconstructing a Tucker model from two-way factors

Computing the Tucker decomposition of a tensor is rather costly in terms of time and memory requirements. Moreover, the decomposition is not unique: the core tensor $\mathcal{G}$ can be modified without affecting the model's fit by applying the inverse modification to the factor matrices. These two drawbacks led us to consider an alternative method for the construction of the Tucker model. Specifically, we consider the factor matrices as given (as the output from our first step), and proceed to compute the core tensor $\mathcal{G}$. Additionally, we do not use a latent representation for the first mode, which means that the first mode is represented by its original instances.

Our model can be straightforwardly applied to language data. The core tensor $\mathcal{G}$ models the latent interactions between verbs, subject, and objects. $\mathcal{G}$ is computed by applying the n-mode product to the appropriate mode of the original tensor (equation 7),

$$\mathcal{G} = \mathcal{X} \times_2 \mathbf{W}^T \times_3 \mathbf{W}^T \quad (7)$$

where $\mathcal{X}^{V \times N \times N}$ is our original data tensor, consisting of the weighted co-occurrence frequencies of *svo* triples (extracted from corpus data), and $\mathbf{W}^{N \times K}$ is our latent factor matrix for nouns. Note that we do not use a latent representation for the verb mode. To be able to efficiently compute the similarity of verbs (both within and outside of compositional phrases), only the *subject* and *object* mode are represented by latent factors, while the *verb* mode is represented by its original instances. This means that our core tensor $\mathcal{G}$ will be of size $V \times K \times K$.[3] A graphical representation is given in figure 3.

Note that both tensor $\mathcal{X}$ and factor matrices $\mathbf{W}$ are non-negative, which means our core tensor $\mathcal{G}$ will also be non-negative.

3.4 The composition of *svo* triples

In order to compute the composition of a particular subject verb object triple $\langle s, v, o \rangle$, we first extract the appropriate subject vector $\mathbf{w}_s$ and object vector $\mathbf{w}_o$ (both of length K) from our factor matrix $\mathbf{W}$, and

[3] It is straightforward to also construct a latent factor model for verbs using NMF, and include it in the construction of our core tensor; we believe such a model might have interesting applications, but we save this as an exploration for future work.

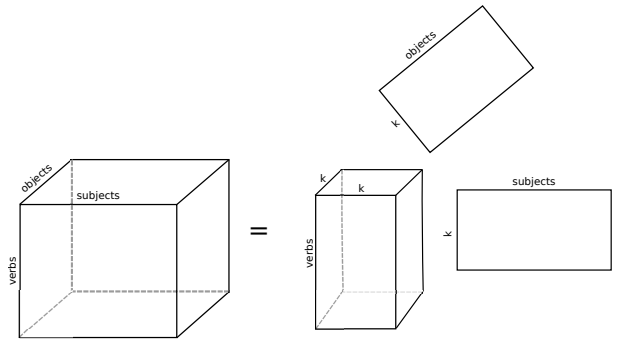

Figure 3: A graphical representation of our model instantiation without the latent verb mode

compute the outer product of both vectors, resulting in a matrix $\mathbf{Y}$ of size $K \times K$.

$$\mathbf{Y} = \mathbf{w}_s \circ \mathbf{w}_o \tag{8}$$

Our second and final step is then to weight the original verb matrix $\mathbf{G}_v$ of latent interactions (the appropriate verb slice of tensor $\mathcal{G}$) with matrix $\mathbf{Y}$, containing the latent interactions of the specific subject and object. This is carried out by taking the Hadamard product of $\mathbf{G}_v$ and $\mathbf{Y}$.

$$\mathbf{Z} = \mathbf{G}_v * \mathbf{Y} \tag{9}$$

4 Example

In this section, we present a number of example computations that clarify how our model is able to capture compositionality. All examples come from actual corpus data, and are computed in a fully automatic and unsupervised way.

Consider the following two sentences:

(1) The athlete runs a race.

(2) The user runs a command.

Both sentences contain the verb *run*, but they represent clearly different actions. When we compute the composition of both instances of *run* with their respective subject and object, we want our model to show this difference.

To compute the compositional representation of sentences (1) and (2), we proceed as follows. First, we extract the latent vectors for subject and object ($\mathbf{w}_{athlete}$ and $\mathbf{w}_{race}$ for the first sentence, $\mathbf{w}_{user}$ and $\mathbf{w}_{command}$ for the second sentence) from matrix $\mathbf{W}$. Next, we compute the outer product of subject and object – $\mathbf{w}_{athlete} \circ \mathbf{w}_{race}$ and $\mathbf{w}_{user} \circ \mathbf{w}_{command}$ – which yields matrices $\mathbf{Y}_{\langle athlete,race \rangle}$ and $\mathbf{Y}_{\langle user,command \rangle}$. By virtue of the outer product, the matrices $\mathbf{Y}$ – of size $K \times K$ – represent the level of interaction between the latent factors of the subject and the latent factors of the object. We can inspect these interactions by looking up the factor pairs (i.e. matrix cells) with the highest values in the matrices $\mathbf{Y}$. Table 1 presents the factor pairs with highest value for matrix $\mathbf{Y}_{\langle athlete,race \rangle}$; table 2 represents the factor pairs with highest value for matrix $\mathbf{Y}_{\langle user,command \rangle}$. In order to render the factors interpretable, we include the three most salient words for the various factors (i.e. the words with the highest value for a particular factor).

The examples in tables 1 and 2 give an impression of the effect of the outer product: semantic features of the subject combine with semantic features of the object, indicating the extent to which these features interact within the expression. In table 1, we notice that animacy features (28, 195) and a sport feature (25) combine with a 'sport event' feature (119). In table 2, we see that similar animacy features (40, 195) and technological features (7, 45) combine with another technological feature (89).

Similarly, we can inspect the latent interactions of the verb *run*, which are represented in the tensor slice $\mathbf{G}_{run}$. Note that this matrix contains the verb semantics computed over the complete corpus. The most salient factor interactions for $\mathbf{G}_{run}$ are represented in table 3.

Table 3 illustrates that different senses of the verb *run* are represented within the matrix $\mathbf{G}_{run}$. The first two factor pairs hint at the 'organize' sense of the verb (*run a seminar*). The third factor pair represents the 'transport' sense of the verb (*the bus runs every hour*).[4] And the fourth factor pair represents the 'execute' or 'deploy' sense of *run* (*run Linux, run a computer program*). Note that we only show the factor pairs with the highest value; matrix $\mathbf{G}$ contains a value for each pairwise combination of the latent factors, effectively representing a rich latent semantics for the verb in question.

The last step is to take the Hadamard product of matrices $\mathbf{Y}$ with verb matrix $\mathbf{G}$, which yields our final

[4] Obviously, *hour* is not an object of the verb, but due to parsing errors it is thus represented.

factors	subject	object	value
$\langle 195,119\rangle$	*people* (.008), *child* (.008), *adolescent* (.007)	*cup* (.007), *championship* (.006), *final* (.005)	.007
$\langle 25,119\rangle$	*hockey* (.007), *poker* (.007), *tennis* (.006)	*cup* (.007), *championship* (.006), *final* (.005)	.004
$\langle 90,119\rangle$	*professionalism* (.007), *teamwork* (.007), *confidence* (.006)	*cup* (.007), *championship* (.006), *final* (.005)	.003
$\langle 28,119\rangle$	*they* (.004), *pupil* (.003), *participant* (.003)	*cup* (.007), *championship* (.006), *final* (.005)	.003

Table 1: Factor pairs with highest value for matrix $\mathbf{Y}_{\langle athlete,race\rangle}$

factors	subject	object	value
$\langle 7,89\rangle$	*password* (.009), *login* (.007), *username* (.007)	*filename* (.007), *null* (.006), *integer* (.006)	.010
$\langle 40,89\rangle$	*anyone* (.004), *reader* (.004), *anybody* (.003)	*filename* (.007), *null* (.006), *integer* (.006)	.007
$\langle 195,89\rangle$	*people* (.008), *child* (.008), *adolescent* (.007)	*filename* (.007), *null* (.006), *integer* (.006)	.006
$\langle 45,89\rangle$	*website* (.004), *Click* (.003), *site* (.003)	*filename* (.007), *null* (.006), *integer* (.006)	.006

Table 2: Factor pairs with highest value for matrix $\mathbf{Y}_{\langle user,command\rangle}$

matrices, $\mathbf{Z}_{run,\langle athlete,race\rangle}$ and $\mathbf{Z}_{run,\langle user,command\rangle}$. The Hadamard product will act as a bidirectional filter on the semantics of both the verb and its subject and object: interactions of semantic features that are present in both matrix $\mathbf{Y}$ and $\mathbf{G}$ will be highlighted, while the other interactions are played down. The result is a representation of the verb's semantics tuned to its particular subject-object combination. Note that this final step can be viewed as an instance of function application (Baroni and Zamparelli, 2010). Also note the similarity to Grefenstette and Sadrzadeh's (2011a,2011b) approach, who equally make use of the elementwise matrix product in order to weight the semantics of the verb.

We can now go back to our original tensor $\mathcal{G}$, and compute the most similar verbs (i.e. the most similar tensor slices) for our newly computed matrices $\mathbf{Z}$.[5]

If we do this for matrix $\mathbf{Z}_{run,\langle athlete,race\rangle}$, our model comes up with verbs *finish* (.29), *attend* (.27), and *win* (.25). If, instead, we compute the most similar verbs for $\mathbf{Z}_{run,\langle user,command\rangle}$, our model yields *execute* (.42), *modify* (.40), *invoke* (.39).

Finally, note that the design of our model naturally takes into account word order. Consider the following two sentences:

(3) man damages car

(4) car damages man

Both sentences contain the exact same words, but the process of damaging described in sentences (3) and (4) is of a rather different nature. Our model is able to take this difference into account: if we compute $\mathbf{Z}_{damage,\langle man,car\rangle}$ following sentence (3), our model yields *crash* (.43), *drive* (.35), *ride* (.35) as most similar verbs. If we do the same for $\mathbf{Z}_{damage,\langle car,man\rangle}$ following sentence (4), our model instead yields *scare* (.26), *kill* (.23), *hurt* (.23).

5 Evaluation

5.1 Methodology

In order to evaluate the performance of our tensor-based factorization model of compositionality, we make use of the sentence similarity task for transitive sentences, defined in Grefenstette and Sadrzadeh (2011a). This is an extension of the similarity task for compositional models developed by Mitchell and Lapata (2008), and constructed according to the same guidelines. The dataset contains 2500 similarity judgements, provided by 25 participants, and is publicly available.[6]

The data consists of transitive verbs, each paired with both a subject and an object noun – thus forming a small transitive sentence. Additionally, a 'landmark' verb is provided. The idea is to compose both the target verb and the landmark verb with subject and noun, in order to form two small compositional

[5] Similarity is calculated by measuring the cosine of the vectorized and normalized representation of the verb matrices.

[6] `http://www.cs.ox.ac.uk/activities/CompDistMeaning/GS2011data.txt`

factors	subject	object	value
$\langle 128,181\rangle$	*Mathematics* (.004), *Science* (.004), *Economics* (.004)	*course* (.005), *tutorial* (.005), *seminar* (.005)	.058
$\langle 293,181\rangle$	*organization* (.007), *association* (.007), *federation* (.006)	*course* (.005), *tutorial* (.005), *seminar* (.005)	.053
$\langle 60,140\rangle$	*rail* (.011), *bus* (.009), *ferry* (.008)	*third* (.004), *decade* (.004), *hour* (.004)	.038
$\langle 268,268\rangle$	*API* (.008), *Apache* (.007), *Unix* (.007)	*API* (.008), *Apache* (.007), *Unix* (.007)	.038

Table 3: Factor combinations for $\mathbf{G}_{run}$

phrases. The system is then required to come up with a suitable similarity score for these phrases. The correlation of the model's judgements with human judgements (scored 1–7) is then calculated using Spearman's ρ. Two examples of the task are provided in table 4.

p	target	subject	object	landmark	sim
19	meet	system	criterion	visit	1
21	write	student	name	spell	6

Table 4: Two example judgements from the phrase similarity task defined by Grefenstette and Sadrzadeh (2011a)

Grefenstette and Sadrzadeh (2011a) seem to calculate the similarity score contextualizing both the target verb and the landmark verb. Another possibility is to contextualize only the target verb, and compute the similarity score with the non-contextualized landmark verb. In our view, the latter option provides a better assessment of the model's similarity judgements, since contextualizing low-similarity landmarks often yields non-sensical phrases (e.g. *system visits criterion*). We provide scores for both contextualized and non-contextualized landmarks.

We compare our results to a number of different models. The first is Mitchell and Lapata's (2008) model, which computes the elementwise vector multiplication of verb, subject and object. The second is Grefenstette and Sadrzadeh's (2011b) best scoring model instantiation of the categorical distributional compositional model (Coecke et al., 2010). This model computes the outer product of the subject and object vector, the outer product of the verb vector with itself, and finally the elementwise product of both results. It yields the best score on the transitive sentence similarity task reported to date.

As a baseline, we compute the non-contextualized similarity score for target verb and landmark. The upper bound is provided by Grefenstette and Sadrzadeh (2011a), based on interannotator agreement.

5.2 Implementational details

All models have been constructed using the UKWAC corpus (Baroni et al., 2009), a 2 billion word corpus automatically harvested from the web. From this data, we accumulate the input matrix $\mathbf{V}$ for our first NMF step. We use the 10K most frequent nouns, cross-classified by the 2K most frequent context words.[7] Matrix $\mathbf{V}$ is weighted using pointwise mutual information (PMI, Church and Hanks (1990)).

A parsed version of the corpus is available, which has been parsed with MaltParser (Nivre et al., 2006). We use this version in order to extract our *svo* triples. From these triples, we construct our tensor $\mathcal{X}$, using 1K verbs $\times$ 10K subjects $\times$ 10K objects. Note once again that the subject and object instances in the second step are exactly the same as the noun instances in the first step. Tensor $\mathcal{X}$ has been weighted using a three-way extension of PMI, following equation 10 (Van de Cruys, 2011).

$$pmi3(x,y,z) = \log \frac{p(x,y,z)}{p(x)p(y)p(z)} \qquad (10)$$

We set $K = 300$ as our number of latent factors. The value was chosen as a trade-off between a model that is both rich enough, and does not require an excessive amount of memory (for the modeling of the core tensor). The algorithm runs fairly efficiently. Each NMF step is computed in a matter of seconds, with convergence after 50–100 iterations. The construction of the core tensor is somewhat more

[7] We use a context window of 5 words, both before and after the target word; a stop list was used to filter out grammatical function words.

evolved, but does not exceed a wall time of 30 minutes. Results have been computed on a machine with Intel Xeon 2.93Ghz CPU and 32GB of RAM.

5.3 Results

The results of the various models are presented in table 5; *multiplicative* represents Mitchell and Lapata's (2008) multiplicative model, *categorical* represents Grefenstette and Sadrzadeh's (2011b) model, and *latent* represents the model presented in this paper.

model	contextualized	non-contextualized
baseline	.23	
multiplicative	**.32**	.34
categorical	**.32**	.35
latent	**.32**	**.37**
upper bound	.62	

Table 5: Results of the different compositionality models on the phrase similarity task

In the contextualized version of the similarity task (in which the landmark is combined with subject and object), all three models obtain the same result (.32). However, in the non-contextualized version (in which only the target verb is combined with subject and object), the models differ in performance. These differences are statistically significant.[8] As mentioned before, we believe the non-contextualized version of the task gives a better impression of the systems' ability to capture compositionality. The contextualization of the landmark verb often yields non-sensical combinations, such as *system visits criterion*. We therefore deem it preferable to compute the similarity of the target verb in composition (*system meets criterion*) to the non-contextualized semantics of the landmark verb (*visit*).

Note that the scores presented in this evaluation (including the baseline score) are significantly higher than the scores presented in Grefenstette and Sadrzadeh (2011b). This is not surprising, since the corpus we use – UKWAC – is an order of magnitude larger than the corpus used in their research – the British National Corpus (BNC). Presumably, the scores are also favoured by our weighting measure. In our experience, PMI performs better than weighting with conditional probabilities.[9]

6 Conclusion

In this paper, we presented a novel method for the computation of compositionality within a distributional framework. The key idea is that compositionality is modeled as a multi-way interaction between latent factors, which are automatically constructed from corpus data. We used our method to model the composition of *subject verb object* combinations. The method consists of two steps. First, we compute a latent factor model for nouns from standard co-occurrence data. Next, the latent factors are used to induce a latent model of three-way *subject verb object* interactions, represented by a core tensor. Our model has been evaluated on a similarity task for transitive phrases, in which it matches and even exceeds the state of the art.

We conclude with a number of future work issues. First of all, we would like to extend our framework in order to incorporate more compositional phenomena. Our current model is designed to deal with the latent modeling of *subject verb object* combinations. We would like to investigate how other compositional phenomena might fit within our latent interaction framework, and how our model is able to tackle the computation of compositionality across a differing number of modes.

Secondly, we would like to further explore the possibilities of our model in which all three modes are represented by latent factors. The instantiation of our model presented in this paper has two latent modes, using the original instances of the verb mode in order to efficiently compute verb similarity. We think a full-blown latent interaction model might prove to have interesting applications in a number of NLP tasks, such as the paraphrasing of compositional expressions.

Finally, we would like to test our method using a number of different evaluation frameworks. We think tasks of similarity judgement have their merits, but in a way are also somewhat limited. In our opinion, research on the modeling of compositional phenomena within a distributional framework would substantially

[8] $p < 0.01$; model differences have been tested using stratified shuffling (Yeh, 2000).

[9] Contrary to the findings of Mitchell and Lapata (2008), who report a high correlation with human similarity judgements.

benefit from new evaluation frameworks. In particular, we think of a lexical substitution or paraphrasing task along the lines of McCarthy and Navigli (2009), but specifically aimed at the assessment of compositional phenomena.

Acknowledgements

Tim Van de Cruys and Thierry Poibeau are supported by the *Centre National de la Recherche Scientifique* (CNRS, France), Anna Korhonen is supported by the *Royal Society* (UK).

References

Brett W. Bader, Tamara G. Kolda, et al. 2012. Matlab tensor toolbox version 2.5. `http://www.sandia.gov/~tgkolda/TensorToolbox/`.

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, Cambridge, MA, October. Association for Computational Linguistics.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556, Jeju Island, Korea, July. Association for Computational Linguistics.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

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

Stephen Clark and Stephen Pulman. 2007. Combining symbolic and distributional models of meaning. In *Proceedings of the AAAI Spring Symposium on Quantum Interaction*, pages 52–55.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributed model of meaning. *Lambek Festschrift, Linguistic Analysis, vol. 36*, 36.

Lieven De Lathauwer, Bart De Moor, and Joseph Vandewalle. 2000. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278.

Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1162–1172, Cambridge, MA, October.

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

Katrin Erk and Sebastian Padó. 2009. Paraphrase assessment in structured vector space: Exploring parameters and datasets. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 57–65, Athens, Greece.

Gottlob Frege. 1892. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50.

Eugenie Giesbrecht. 2010. Towards a matrix-based distributional model of meaning. In *Proceedings of the NAACL HLT 2010 Student Research Workshop*, pages 23–28. Association for Computational Linguistics.

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

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Richard A Harshman and Margaret E Lundy. 1994. Parafac: Parallel factor analysis. *Computational Statistics & Data Analysis*, 18(1):39–72.

Tamara G. Kolda and Brett W. Bader. 2009. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September.

Tamara G. Kolda and Jimeng Sun. 2008. Scalable tensor decompositions for multi-aspect data mining. In *ICDM 2008: Proceedings of the 8th IEEE International Conference on Data Mining*, pages 363–372, December.

Thomas Landauer and Susan Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis the-

ory of the acquisition, induction, and representation of knowledge. *Psychology Review*, 104:211–240.

Daniel D. Lee and H. Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562.

Diana McCarthy and Roberto Navigli. 2009. The English lexical substitution task. *Language resources and evaluation*, 43(2):139–159.

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

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC-2006*, pages 2216–2219.

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, Jeju Island, Korea, July. Association for Computational Linguistics.

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

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

Ledyard R. Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.

Tim Van de Cruys. 2010. A non-negative tensor factorization model for selectional preference induction. *Natural Language Engineering*, 16(4):417–437.

Tim Van de Cruys. 2011. Two multivariate generalizations of pointwise mutual information. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, pages 16–20, Portland, Oregon, USA, June. Association for Computational Linguistics.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics*, pages 947–953, Saarbrücken, Germany.