Estimating Linear Models for Compositional Distributional Semantics

Fabio Massimo Zanzotto[1]
(1) Department of Computer Science
University of Rome "Tor Vergata"
zanzotto@info.uniroma2.it

Ioannis Korkontzelos
Department of Computer Science
University of York
johnkork@cs.york.ac.uk

Francesca Fallucchi[1,2]
(2) Università Telematica
"G. Marconi"
f.fallucchi@unimarconi.it

Suresh Manandhar
Department of Computer Science
University of York
suresh@cs.york.ac.uk

Abstract

In distributional semantics studies, there is a growing attention in compositionally determining the distributional meaning of word sequences. Yet, compositional distributional models depend on a large set of parameters that have not been explored. In this paper we propose a novel approach to estimate parameters for a class of compositional distributional models: the additive models. Our approach leverages on two main ideas. Firstly, a novel idea for extracting compositional distributional semantics examples. Secondly, an estimation method based on regression models for multiple dependent variables. Experiments demonstrate that our approach outperforms existing methods for determining a good model for compositional distributional semantics.

1 Introduction

Lexical distributional semantics has been largely used to model word meaning in many fields as computational linguistics (McCarthy and Carroll, 2003; Manning et al., 2008), linguistics (Harris, 1964), corpus linguistics (Firth, 1957), and cognitive research (Miller and Charles, 1991). The fundamental hypothesis is the distributional hypothesis (DH): "similar words share similar contexts" (Harris, 1964). Recently, this hypothesis has been operationally defined in many ways in the fields of physicology, computational linguistics, and information retrieval (Li et al., 2000; Pado and Lapata, 2007; Deerwester et al., 1990).

Given the successful application to words, distributional semantics has been extended to word sequences. This has happened in two ways: (1) via the reformulation of DH for specific word sequences (Lin and Pantel, 2001); and (2) via the definition of compositional distributional semantics (CDS) models (Mitchell and Lapata, 2008; Jones and Mewhort, 2007). These are two different ways of addressing the problem.

Lin and Pantel (2001) propose the *pattern distributional hypothesis* that extends the distributional hypothesis for specific patterns, i.e. word sequences representing partial verb phrases. Distributional meaning for these patterns is derived directly by looking to their occurrences in a corpus. Due to data sparsity, patterns of different length appear with very different frequencies in the corpus, affecting their statistics detrimentally. On the other hand, compositional distributional semantics (CDS) propose to obtain distributional meaning for sequences by composing the vectors of the words in the sequences (Mitchell and Lapata, 2008; Jones and Mewhort, 2007). This approach is fairly interesting as the distributional meaning of sequences of different length is obtained by composing distributional vectors of single words. Yet, many of these approaches have a large number of parameters that cannot be easily estimated.

In this paper we propose a novel approach to es-

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 1263–1271,
Beijing, August 2010

timate parameters for additive compositional distributional semantics models. Our approach leverages on two main ideas. Firstly, a novel way for extracting compositional distributional semantics examples and counter-examples. Secondly, an estimation model that exploits these examples and determines an equation system that represents a regression problem with multiple dependent variables. We propose a method to estimate a solution of this equation system based on the Moore-Penrose pseudo-inverse matrices (Penrose, 1955).

The rest of the paper is organised as follows: Firstly, we shortly review existing compositional distributional semantics (CDS) models (Sec. 2). Then we describe our model for estimating CDS models parameters (Sec. 3). In succession, we introduce a way to extract compositional distributional semantics examples from dictionaries (Sec. 4). Then, we discuss the experimental set up and the results of our linear CDS model with estimated parameters with respect to existing CDS models (Sec. 5).

2 Models for compositional distributional semantics (CDS)

A CDS model is a function $\odot$ that computes the distributional vector of a sequence of words s by combining the distributional vectors of its component words $\mathsf{w}_1 \dots \mathsf{w}_n$. Let $\odot(\mathsf{s})$ be the distributional vector describing s and $\vec{w}_i$ the distributional vectors describing its component word w_i. Then, the CDS model can be written as:

$$\odot(\mathsf{s}) = \odot(\mathsf{w}_1 \dots \mathsf{w}_n) = \vec{w}_1 \odot \dots \odot \vec{w}_n \qquad (1)$$

This generic model has been fairly studied and many different functions have been proposed and tested.

Mitchell and Lapata (2008) propose the following general CDS model for 2-word sequences $\mathsf{s} = \mathsf{xy}$:

$$\odot(\mathsf{s}) = \odot(\mathsf{xy}) = f(\vec{x}, \vec{y}, R, K) \qquad (2)$$

where $\vec{x}$ and $\vec{y}$ are respectively the distributional vectors of x and y, R is the particular syntactic and/or semantic relation connecting x and y, and, K represents the amount of background knowledge that the vector composition process takes into account. Two specialisations of the general CDS model are proposed: the *basic additive* model and the *basic multiplicative* model.

	vector dimensions				
	between	gap	process	social	two
contact	< 11,	0,	3,	0,	11 >
x: *close*	< 27,	3,	2,	5,	24 >
y: *interaction*	< 23,	0,	3,	8,	4 >

Table 1: Example of distributional frequency vectors for the triple $t = (\vec{contact}, \vec{close}, \vec{interaction})$

The *basic additive* model (BAM) is written as:

$$\odot(\mathsf{s}) = \alpha\vec{x} + \beta\vec{y} \qquad (3)$$

where α and β are two scalar parameters. The simplistic parametrisation is $\alpha = \beta = 1$. For example, given the vectors $\vec{x}$ and $\vec{y}$ of Table 1, $\odot_{BAM}(\mathsf{s}) =< 50, 3, 5, 13, 28 >$.

The *basic multiplicative* model (BMM) is written as:

$$s_i = x_i y_i \qquad (4)$$

where s_i, x_i, and y_i are the i-th dimensions of the vectors $\odot(\mathsf{s})$, $\vec{x}$, and $\vec{y}$, respectively. For the example of Table 1, $\odot_{BMM}(\mathsf{s}) =< 621, 0, 6, 40, 96 >$.

Erk and Padó (2008) look at the problem in a different way. Let the general distributional meaning of the word w be $\vec{w}$. Their model computes a different vector $\vec{w}_{\mathsf{s}}$ that represents the specific distributional meaning of w with respect to s, i.e.:

$$\vec{w}_{\mathsf{s}} = \oslash(\mathsf{w}, \mathsf{s}) \qquad (5)$$

In general, this operator gives different vectors for each word w_i in the sequence s, i.e. $\oslash(\mathsf{w}_i, \mathsf{s}) \neq \oslash(\mathsf{w}_j, \mathsf{s})$ if $i \neq j$. It also gives different vectors for a word w_i appearing in different sequences s_k and s_l, i.e. $\oslash(\mathsf{w}_i, \mathsf{s}_k) \neq \oslash(\mathsf{w}_i, \mathsf{s}_l)$ if $k \neq l$.

The model of Erk and Padó (2008) was designed to *disambiguate* the distributional meaning of a word w in the context of the sequence s. However, substituting the word w with the semantic head h of s, allows to compute the distributional meaning of sequence s as shaped by the word that is governing the sequence (c.f. Pollard and Sag (1994)). For example, the distributional meaning of the word sequence *eats mice* is governed by the verb *eats*. Following this model, the distributional vector $\odot(\mathsf{s})$ can be written as:

$$\odot(\mathsf{s}) \approx \oslash(\mathsf{h}, \mathsf{s}) \qquad (6)$$

The function $\oslash(\mathsf{h}, \mathsf{s})$ explicitly uses the relation R and the knowledge K of the general equation 2, being based on the notion of selectional preferences. We exploit the model for sequences of two words s=xy where the two words are related with an oriented syntactic relation r (e.g. r=adj_modifier). For making the syntactic relation explicit, we indicate the sequence as: $\mathsf{s} = \mathsf{x} \overset{r}{\leftarrow} \mathsf{y}$.

Given a word w, the model has to keep track of its selectional preferences. Consequently, each word w is represented with a triple:

$$(\vec{w}, R_{\mathsf{w}}, R_{\mathsf{w}}^{-1}) \qquad (7)$$

where $\vec{w}$ is the distributional vector of the word w, R_{w} is the set of the vectors representing the direct selectional preferences of the word w, and R_{w}^{-1} is the set of the vectors representing the indirect selectional preferences of the word w. Given a set of syntactic relations $\mathcal{R}$, the set R_{w} and R_{w}^{-1} contain respectively a selectional preference vector $R_{\mathsf{w}}(r)$ and $R_{\mathsf{w}}(r)^{-1}$ for each $r \in \mathcal{R}$. Selectional preferences are computed as in Erk (2007). If x is the semantic head of sequence s, then the model can be written as:

$$\odot(\mathsf{s}) = \oslash(\mathsf{x}, \mathsf{x} \overset{r}{\leftarrow} \mathsf{y}) = \vec{x} \odot R_{\mathsf{y}}(r) \qquad (8)$$

Otherwise, if y is the semantic head:

$$\odot(\mathsf{s}) = \oslash(\mathsf{y}, \mathsf{x} \overset{r}{\leftarrow} \mathsf{y}) = \vec{y} \odot R_{\mathsf{x}}^{-1}(r) \qquad (9)$$

$\odot$ is in both cases realised using BAM or BMM. We will call these models: *basic additive* model with selectional preferences (BAM-SP) and basic multiplicative model with selectional preferences (BMM-SP).

Both Mitchell and Lapata (2008) and Erk and Padó (2008) experimented with few empirically estimated parameters. Thus, the general additive CDS model has not been adequately explored.

3 Estimating Additive Compositional Semantics Models from Data

The generic *additive* model sums the vectors $\vec{x}$ and $\vec{y}$ in a new vector $\vec{z}$:

$$\odot(\mathsf{s}) = \vec{z} = A\vec{x} + B\vec{y} \qquad (10)$$

where A and B are two square matrices capturing the relation R and the background knowledge K of equation 2. Writing matrices A and B by hand is impossible because of their large size. Estimating these matrices is neither a simple classification learning problem nor a simple regression problem. It is a regression problem with multiple dependent variables. In this section, we propose our model to solve this regression problem using a set of training examples E.

The set of training examples E contains triples of vectors $(\vec{z}, \vec{x}, \vec{y})$. $\vec{x}$ and $\vec{y}$ are the two distributional vectors of the words x and y. $\vec{z}$ is the expected distributional vector of the composition of $\vec{x}$ and $\vec{y}$. Note that for an ideal perfectly performing CDS model we can write $\vec{z} = \odot(\mathsf{xy})$. However, in general the expected vector $\vec{z}$ is not guaranteed to be equal to the composed one $\odot(\mathsf{xy})$. Figure 1 reports an example of these triples, i.e., $t = (\vec{contact}, \vec{close}, \vec{interaction})$, with the related distributional vectors. The construction of E is discussed in section 4.

In the rest of the section, we describe how the regression problem with multiple dependent variables can be solved with a linear equation system and we give a possible solution of this equation system. In the experimental section, we refer to our model as the estimated additive model (EAM).

3.1 Setting the linear equation system

The matrices A and B of equation 10 can be joined in a single matrix:

$$\vec{z} = \begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix} \qquad (11)$$

For the triple t of table 1, equation 11 is:

$$\vec{contact} = \begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} \vec{close} \\ \vec{interaction} \end{pmatrix} \qquad (12)$$

and it can be rewritten as:

$$\begin{pmatrix}11\\0\\3\\0\\11\end{pmatrix} = \begin{pmatrix}A_{5\times5} & B_{5\times5}\end{pmatrix}\begin{pmatrix}27\\3\\2\\5\\24\\23\\0\\3\\8\\4\end{pmatrix} \tag{13}$$

Focusing on matrix $\begin{pmatrix}AB\end{pmatrix}$, we can transpose the matrices as follows:

$$\begin{aligned}\vec{z}^T &= \left(\begin{pmatrix}A & B\end{pmatrix}\begin{pmatrix}\vec{x}\\\vec{y}\end{pmatrix}\right)^T\\ &= \begin{pmatrix}\vec{x}^T & \vec{y}^T\end{pmatrix}\begin{pmatrix}A^T\\B^T\end{pmatrix}\end{aligned} \tag{14}$$

Matrix $\begin{pmatrix}\vec{x}^T & \vec{y}^T\end{pmatrix}$ is known and matrix $\begin{pmatrix}A^T\\B^T\end{pmatrix}$ is to be estimated.

Equation 14 is the prototype of our final equation system. The larger the matrix $\begin{pmatrix}AB\end{pmatrix}$ to be estimated, the more equations like 14 are needed. Given set E that contains n triples $(\vec{z}, \vec{x}, \vec{y})$, we can write the following system of equations:

$$\begin{pmatrix}\vec{z}_1^T\\\vec{z}_2^T\\\vdots\\\vec{z}_n^T\end{pmatrix} = \begin{pmatrix}\begin{pmatrix}\vec{x}_1^T & \vec{y}_1^T\end{pmatrix}\\\begin{pmatrix}\vec{x}_2^T & \vec{y}_2^T\end{pmatrix}\\\vdots\\\begin{pmatrix}\vec{x}_n^T & \vec{y}_n^T\end{pmatrix}\end{pmatrix}\begin{pmatrix}A^T\\B^T\end{pmatrix} \tag{15}$$

The vectors derived from the triples can be seen as two matrices of n rows, Z and $\begin{pmatrix}XY\end{pmatrix}$ related to $\vec{z}_i^T$ and $\begin{pmatrix}\vec{x}_i^T & \vec{y}_i^T\end{pmatrix}$, respectively. The overall equation system is then the following:

$$Z = \begin{pmatrix}X & Y\end{pmatrix}\begin{pmatrix}A^T\\B^T\end{pmatrix} \tag{16}$$

This equation system represents the constraints that matrices A and B have to satisfy in order to be a possible linear CDS model that can at least describe seen examples. We will hereafter call $\Lambda = \begin{pmatrix}A & B\end{pmatrix}$ and $Q = \begin{pmatrix}X & Y\end{pmatrix}$. The system in equation 16 can be simplified as:

$$Z = Q\Lambda^T \tag{17}$$

As Q is a rectangular and singular matrix, it is not invertible and the system in equation 16 has no solutions. It is possible to use the principle of Least Square Estimation for computing an approximation solution. The idea is to compute the solution $\widehat{\Lambda}$ that minimises the residual norm, i.e.:

$$\widehat{\Lambda}^T = \arg\min_{\Lambda^T} \|Q\Lambda^T - Z\|^2 \tag{18}$$

One solution for this problem is the **Moore-Penrose pseudoinverse** Q^+ (Penrose, 1955) that gives the following final equation:

$$\widehat{\Lambda}^T = Q^+ Z \tag{19}$$

In the next section, we discuss how the **Moore-Penrose pseudoinverse** is obtained using singular value decomposition (SVD).

3.2 Computing the pseudo-inverse matrix

The pseudo-inverse matrix can provide an approximated solution even if the equation system has no solutions. We here compute the **Moore-Penrose pseudoinverse** using singular value decomposition (SVD) that is widely used in computational linguistics and information retrieval for reducing spaces (Deerwester et al., 1990).

Moore-Penrose pseudoinverse (Penrose, 1955) is computed in the following way. Let the original matrix Q have n rows and m columns and be of rank r. The SVD decomposition of the original matrix Q is $Q = U\Sigma V^T$ where Σ is a square diagonal matrix of dimension r. Then, the pseudo-inverse matrix that minimises the equation 18 is:

$$Q^+ = V\Sigma^+ U^T \tag{20}$$

where the diagonal matrix Σ^+ is the $r \times r$ transposed matrix of Σ having as diagonal elements the reciprocals of the singular values $\frac{1}{\delta_1}, \frac{1}{\delta_2}, ..., \frac{1}{\delta_r}$ of Σ.

Using SVD to compute the pseudo-inverse matrix allows for different approximations (Fallucchi and Zanzotto, 2009). The algorithm for computing the singular value decomposition is iterative (Golub and Kahan, 1965). Firstly derived dimensions have higher singular value. Then, dimension k is more informative than dimension $k' > k$. We can consider different values for k to obtain different SVD for the approximations Q_k^+ of the original matrix Q^+ in equation 20), i.e.:

$$Q_k^+ = V_{n\times k}\Sigma_{k\times k}^+ U_{k\times m}^T \tag{21}$$

where Q_k^+ is a matrix n by m obtained considering the first k singular values.

4 Building positive and negative examples

As explained in the previous section, estimating CDS models, needs a set of triples E, similar to triple t of table 1. This set E should contain positive examples in the form of triples $(\vec{z}_i, \vec{x}_i, \vec{y}_i)$. Examples are positive in the sense that $\vec{z}_i = \odot(\mathsf{xy})$ for an ideal CDS. There are no available sets to contain such triples, with the exception of the set used in Mitchell and Lapata (2008) which is designed only for testing purposes. It contains similar and dissimilar pairs of sequences (s_1,s_2) where each sequence is a verb-noun pair (v_i,n_i). From the positive part of this set, we can only derive quadruples where $\odot(\mathsf{v}_1\mathsf{n}_1) \approx \odot(\mathsf{v}_2\mathsf{n}_2)$ but we cannot derive the ideal resulting vector of the composition $\odot(\mathsf{v}_i\mathsf{n}_i)$. Sets used to test multiword expression (MWE) detection models (e.g., (Schone and Jurafsky, 2001; Nicholson and Baldwin, 2008; Kim and Baldwin, 2008; Cook et al., 2008; Villavicencio, 2003; Korkontzelos and Manandhar, 2009)) are again not useful as containing only valid MWE that cannot be used to determine the set of training triples needed here.

As a result, we need a novel idea to build sets of triples to train CDS models. We can leverage on knowledge stored in dictionaries. In the rest of the section, we describe how we build the positive example set E and a control negative example set NE. Elements of the two sets are pairs (t,s) where t is a target word s is a sequence of words. t is the word that represent the distributional meaning of s in the case of E. Contrarily, t is totally unrelated to the distributional meaning of s in NE. The sets E and NE can be used both for training and for testing. In the testing phase, we can use these sets to determine whether a CDS model is good or not and to compare different CDS models.

4.1 Building Positive Examples using Dictionaries

Dictionaries as natural repositories of equivalent expressions can be used to extract positive examples for training and testing CDS models. The basic idea is the following: dictionary entries are *declarations* of equivalence. Words or, occasionally, multi-word expressions t are declared to be semantically similar to their definition sequences s. This happens at least for some sense of the defined words. We can then observe that $\mathsf{t} \approx \mathsf{s}$. For example, we report some sample definitions of *contact* and *high life*:

target word (t)	definition sequence (s)
contact	close interaction
high life	excessive spending

In the first case, a word, i.e. *contact*, is semantically similar to a two-word expression, i.e. *close interaction*. In the second case, two two-word expressions are semantically similar.

Then, the pairs (t, s) can be used to model positive cases of compositional distributional semantics as we know that the word sequence s is compositional and it describes the meaning of the word t. The distributional meaning $\vec{t}$ of t is the expected distributional meaning of s. Consequently, the vector $\vec{t}$ is what the CDS model $\odot(\mathsf{s})$ should compositionally obtain from the vectors of the components $\vec{s_1} \dots \vec{s_m}$ of s. This way of extracting similar expressions has some interesting properties:

First property Defined words t are generally single words. Thus, we can extract stable and meaningful distributional vectors for these words and then compare them to the distributional vectors composed by CDS model. This is an important property as we cannot compare directly the distributional vector $\vec{s}$ of a word sequence s and the vector $\odot(\mathsf{s})$ obtained by composing its components. As the word sequence s grows in length, the reliability of the vector $\vec{s}$ decreases since the sequence s becomes rarer.

Second property Definitions s have a large variety of different syntactic structures ranging from simple structures as Adjective-Noun to more complex ones. This gives the possibility to train and test CDS models that take into account syntax. Table 2 represents the distribution of the more frequent syntactic structures in the definitions of WordNet[1] (Miller, 1995).

[1] Definitions were extracted from WordNet 3.0 and were parsed with the Charniak parser (Charniak, 2000)

Freq.	*Structure*
2635	(FRAG (PP (IN) (NP (DT) (JJ) (NN))))
833	(NP (DT) (JJ) (NN))
811	(NP (NNS))
645	(NP (NNP))
623	(S (VP (VB) (ADVP (RB))))
610	(NP (JJ) (NN))
595	(NP (NP (DT) (NN)) (PP (IN) (NP (NN))))
478	(NP (NP (DT) (NN)) (PP (IN) (NP (NNP))))
451	(FRAG (PP (IN) (NP (NN))))
419	(FRAG (RB) (ADJP (JJ)))
375	(S (VP (VB) (PP (IN) (NP (DT) (NN)))))
363	(S (VP (VB) (PP (IN) (NP (NN)))))
342	(NP (NP (DT) (NN)) (PP (IN) (NP (DT) (NN))))
341	(NP (DT) (JJ) (JJ) (NN))
330	(ADJP (RB) (JJ))
307	(NP (JJ) (NNS))
244	(NP (DT) (NN) (NN))
241	(S (NP (NN)) (NP (NP (NNS)) (PP (IN) (NP (DT) (NNP)))))
239	(NP (NP (DT) (JJ) (NN)) (PP (IN) (NP (DT) (NN))))

Table 2: Top 20 syntactic structures of WordNet definitions

4.2 Extracting Negative Examples from Word Etymology

In order to devise complete training and testing sets for CDS models, we need to find a sensible way to extract negative examples. An option is to randomly generate totally unrelated triples for the negative examples set, NE. In this case, due to data sparseness NE would mostly contain triples $(\vec{z}, \vec{x}, \vec{y})$ where it is expected that $\vec{z} \neq \odot(\mathsf{xy})$. Yet, these can be too generic and too loosely related to be interesting cases.

Instead we attempt to extract sets of negative pairs (t,s) comparable with the one used for building the training set E. The target word t should be a single word and s should be a sequence of words. The latter should be a sequence of words related by construction to t but the meaning of t and s should be unrelated.

The idea is the following: many words are etymologically derived from very old or ancient words. These words represent a collocation which is in general not related to the meaning of the target word. For example, the word *philosophy* derives from two Greek words *philos* (beloved) and *sophia* (wisdom). However, the use of the word *philosophy* in not related to the collocation *beloved wisdom*. This word has lost its original compositional meaning. The following table shows some more etymologically complex words along with the compositionally unrelated collocations:

target word	compositionally unrelated seq.
municipal	receive duty
octopus	eight foot

As the examples suggest, we are able to build a set NE with features similar to the features of N. In particular, each target word is paired with a related word sequence derived from its etymology. These etymologically complex words are unrelated to the corresponding compositional collocations. To derive a set NE with the above characteristics we can use dictionaries containing etymological information as Wiktionary[2].

5 Experimental evaluation

In the previous sections, we presented the estimated additive model (EAM): our approach to estimate the parameters of a generic additive model for CDS. In this section, we experiment with this model to determine whether it performs better than existing models: the basic additive model (BAM), the basic multiplicative model (BMM), the basic additive model with selectional preferences (BAM-SP), and the basic multiplicative model with selectional preferences (BMM-SP) (c.f. Sec. 2). In succession, we explore whether our estimated additive model (EAM) is better than any possible BAM obtained with parameter adjustment. In the rest of the section, we firstly give the experimental setup and then we discuss the experiments and the results.

5.1 Experimental setup

Our experiments aim to compare compositional distributional semantic (CDS) models $\odot$ with respect to their ability of detecting statistically significant difference between sets E and NE. In particular, the average similarity $sim(\vec{z}, \odot(\mathsf{xy}))$ for $(\vec{z}, \vec{x}, \vec{y}) \in E$ should be significantly different from $sim(\vec{z}, \odot(\mathsf{xy}))$ for $(\vec{z}, \vec{x}, \vec{y}) \in NE$. In this section, we describe the chosen similarity measure sim, statistical significance testing and construction details for the training and testing set.

Cosine similarity was used to compare the context vector $\vec{z}$ representing the target word z with the composed vector $\odot(\mathsf{xy})$ representing the context vector of sequence x y. Cosine similarity be-

[2] http://www.wiktionary.org

tween two vectors $\vec{x}$ and $\vec{y}$ of the same dimension is defined as:

$$sim(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \ \|\vec{y}\|} \qquad (22)$$

where $\cdot$ is the dot product and $\|\vec{a}\|$ is the magnitude of vector $\vec{a}$ computed the Euclidean norm.

To evaluate whether a CDS model distinguishes positive examples E from negative examples NE, we test if the distribution of similarities $sim(\vec{z}, \odot(\mathsf{xy}))$ for $(\vec{z}, \vec{x}, \vec{y}) \in E$ is statistically different from the distribution of the same similarities for $(\vec{z}, \vec{x}, \vec{y}) \in NE$. For this purpose, we used Student's t-test for two independent samples of different sizes. t-test assumes that the two distributions are Gaussian and determines the probability that they are similar, i.e., derive from the same underlying distribution. Low probabilities indicate that the distributions are highly dissimilar and that the corresponding CDS model performs well, as it detects statistically different similarities for the positive set E and the negative set NE.

Based on the null hypothesis that the means of the two samples are equal, $\mu_1 = \mu_2$, Student's t-test takes into account the sizes N, means M and variances s^2 of the two samples to compute the following value:

$$t = (M_1 - M_2) \sqrt[-1]{\frac{2(s_1^2 + s_2^2)}{df * N_h}} \qquad (23)$$

where $df = N_1 + N_2 - 2$ stands for the degrees of freedom and $N_h = 2(N_1^{-1} + N_2^{-1})^{-1}$ is the harmonic mean of the sample sizes. Given the statistic t and the degrees of freedom df, we can compute the corresponding p-value, i.e., the probability that the two samples derive from the same distribution. The null hypothesis can be rejected if the p-value is below the chosen threshold of statistical significance (usually 0.1, 0.05 or 0.01), otherwise it is accepted. In our case, rejecting the null hypothesis means that the similarity values of instances of E are significantly different from instances of NE, and that the corresponding CDS model perform well. p-value can be used as a performance ranking function for CDS models.

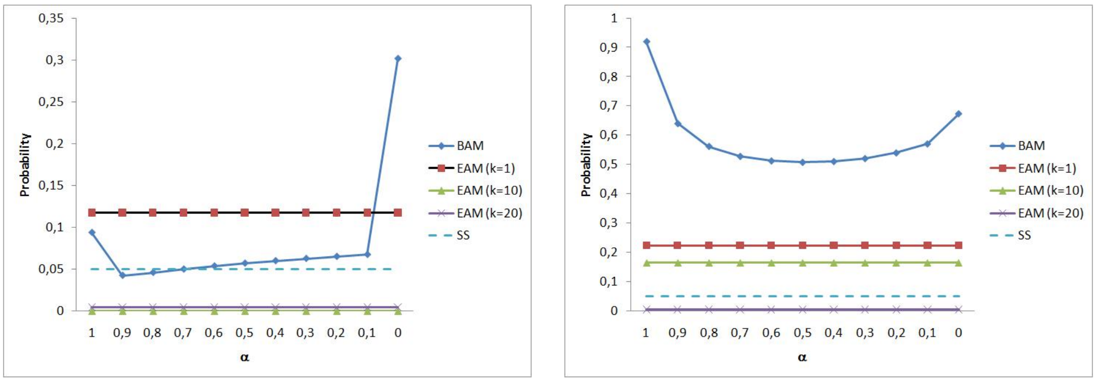
We constructed two sets of instances: (a) a set containing Adjective-Noun or Noun-Noun sequences (*NN* set); and (b) a set containing Verb-Noun sequences (*VN* set). Capturing different syntactic relations, these two sets can support that our results are independent from the syntactic relation between the words of each sequence. For each set, we used WordNet for extracting positive examples E and Wiktionary for extracting negative examples NE as described in Section 4. We obtained the following sets: (a) *NN* consists of 1065 word-sequence pairs from WordNet definitions and 377 pairs extracted from Wiktionary; and (b) *VN* consists of 161 word-sequence pairs from WordNet definitions and 111 pairs extracted from Wiktionary. We have then divided these two sets in two parts of 50% each, for training and testing. Instances of the training part of E have been used to estimate matrices A and B for model EAM, while the testing parts have been used for testing all models. Frequency vectors for all single words occurring in the above pairs were constructed from the British National Corpus using sentences as contextual windows and words as features. The resulting space has 689191 features.

	NN set	*VN* set
BAM	0.05690	0.50753
BMM	0.20262	0.37523
BAM-SP	0.42574	0.01710
BMM-SP	<1.00E-10	0.23552
EAM (k=20)	0.00431	0.00453

Table 3: Probability of confusing E and NE with different CDS models

5.2 Results and Analysis

The first set of experiments compares EAM with other existing CDS models: BAM, BMM, BAM-SP, and BMM-SP. Results are shown in Table 3. The table reports the p-value, i.e., the probability of confusing the positive set E and the negative set NE for all models. Lower probabilities characterise better models. Probabilities below 0.05 indicate that the model detects a statistically significant difference between sets E and NE. EAM has been computed with $k = 20$ different dimensions for the pseudo-inverse matrix. The two basic additive models (BAM and BAM-SP) have been computed for $\alpha = \beta = 1$.



NN set $VN\ set$

Figure 1: p-values of BAM with different values for parameter α (where $\beta = 1 - \alpha$) and of EAM for different approximations of the SVD pseudo-inverse matrix (k)

The first observation is that EAM models significantly separate positive from negative examples for both sets. This is not the case for any of the other models. Only, the selectional preferences based models in two cases have this property, but this cannot be generalised: BAM-SP on the *VN* set and BMM-SP on the *NN* set. In general, these models do not offer the possibility of separating positive from negative examples.

In the second set of experiments, we attempt to investigate whether simple parameter adjustment of BAM can perform better than EAM. Results are shown in figure 1. Plots show the basic additive model (BAM) with different values for parameter α (where $\beta = 1 - \alpha$) and EAM computed for different approximations of the SVD pseudo-inverse matrix (i.e., with different k). The x-axis of the plots represents parameter α and the y-axis represents the probability of confusing the positive set E and the negative set NE. The representation focuses on the performance of BAM with respect to different α values. The performance of EAM for different k values is represented with horizontal lines. Probabilities of different models are directly comparable. Line SS represents the threshold of statistical significance; the value below which the detected difference between the E and NE sets becomes statistically significant.

Experimental results show some interesting facts: While BAM for $\alpha > 0$ perform better than EAM computed with $k = 1$ in the NN set, they do not perform better in the VN set. EAM with $k = 1$ has 1 degree of freedom corresponding to 1 parameter, the same as BAM. The parameter of EAM is tuned on the training set, in contrast to α, the parameter of BAM. Increasing the number of considered dimensions, k of EAM, estimated models outperform BAM for all values of parameter α. Moreover, EAM detect a statistically significant difference between the E and the NE sets for $k \geq 10$ and $k = 20$ for the *NN set* and the *VN set* set, respectively. Simple parametrisation of a BAM does not outperform the proposed estimated additive model.

6 Conclusions

In this paper, we presented an innovative method to estimate linear compositional distributional semantics models. The core of our approach consists on two parts: (1) providing a method to estimate the regression problem with multiple dependent variables and (2) providing a training set derived from dictionary definitions. Experiments showed that our model is highly competitive with respect to state-of-the-art models for compositional distributional semantics.

References

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *proceedings of the 1st NAACL*, pages 132–139, Seattle, Washington.

Cook, Paul, Afsaneh Fazly, and Suzanne Stevenson. 2008. The VNC-Tokens Dataset. In *proceedings of the LREC Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008)*, Marrakech, Morocco.

Deerwester, Scott C., Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Erk, Katrin 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. Association for Computational Linguistics.

Erk, Katrin. 2007. A simple, similarity-based model for selectional preferences. In *proceedings of ACL*. Association for Computer Linguistics.

Fallucchi, Francesca and Fabio Massimo Zanzotto. 2009. SVD feature selection for probabilistic taxonomy learning. In *proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 66–73. Association for Computational Linguistics, Athens, Greece.

Firth, John R. 1957. *Papers in Linguistics.* Oxford University Press, London.

Golub, Gene and William Kahan. 1965. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224.

Harris, Zellig. 1964. Distributional structure. In Katz, Jerrold J. and Jerry A. Fodor, editors, *The Philosophy of Linguistics*, New York. Oxford University Press.

Jones, Michael N. and Douglas J. K. Mewhort. 2007. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114:1–37.

Kim, Su N. and Timothy Baldwin. 2008. Standardised evaluation of english noun compound interpretation. In *proceedings of the LREC Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 39–42, Marrakech, Morocco.

Korkontzelos, Ioannis and Suresh Manandhar. 2009. Detecting compositionality in multi-word expressions. In *proceedings of ACL-IJCNLP 2009*, Singapore.

Li, Ping, Curt Burgess, and Kevin Lund. 2000. The acquisition of word meaning through global lexical co-occurrences. In *proceedings of the 31st Child Language Research Forum.*

Lin, Dekang and Patrick Pantel. 2001. DIRT-discovery of inference rules from text. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD-01).* San Francisco, CA.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval.* Cambridge University Press, Cambridge, UK.

McCarthy, Diana and John Carroll. 2003. Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, 29(4):639–654.

Miller, George A. and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, VI:1–28.

Miller, George A. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

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

Nicholson, Jeremy and Timothy Baldwin. 2008. Interpreting compound nominalisations. In *proceedings of the LREC Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 43–45, Marrakech, Morocco.

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

Penrose, Roger. 1955. A generalized inverse for matrices. In *Proceedings of Cambridge Philosophical Society.*

Pollard, Carl J. and Ivan A. Sag. 1994. *Head-driven Phrase Structured Grammar.* Chicago CSLI, Stanford.

Schone, Patrick and Daniel Jurafsky. 2001. Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In Lee, Lillian and Donna Harman, editors, *proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 100–108.

Villavicencio, Aline. 2003. Verb-particle constructions and lexical resources. In *proceedings of the ACL 2003 workshop on Multiword expressions*, pages 57–64, Morristown, NJ, USA. Association for Computational Linguistics.