Natural Language Engineering **16** (4): 417–437. © Cambridge University Press 2010
doi:10.1017/S1351324910000148

A non-negative tensor factorization model for selectional preference induction

TIM VAN DE CRUYS*
INRIA & Université Paris 7, Rocquencourt, France
e-mail: timvdc@gmail.com

(*Received 15 July 2009; revised 4 March 2010; accepted 14 May 2010*)

Abstract

The distributional similarity methods have proven to be a valuable tool for the induction of semantic similarity. Until now, most algorithms use two-way co-occurrence data to compute the meaning of words. Co-occurrence frequencies, however, need not be pairwise. One can easily imagine situations where it is desirable to investigate co-occurrence frequencies of three modes and beyond. This paper will investigate tensor factorization methods to build a model of three-way co-occurrences. The approach is applied to the problem of selectional preference induction, and automatically evaluated in a pseudo-disambiguation task. The results show that tensor factorization, and non-negative tensor factorization in particular, is a promising tool for Natural Language Processing (NLP).

1 Introduction

The distributional similarity methods have proven to be a valuable tool for the induction of semantic similarity. The aggregate of a word's contexts generally provides enough information to compute its meaning, viz. its semantic similarity or relatedness to other words.

Until now, most algorithms use two-way co-occurrence data to compute the meaning of words. A word's meaning might, for example, be computed by looking at

- the various documents that the word appears in (words × documents);
- a bag of words context window around the word (words × context words);
- the dependency relations that the word appears with (words × dependency relations).

The extracted data – representing the co-occurrence frequencies of two different entities – is encoded in a matrix. The co-occurrence frequencies, however, need not be pairwise. One can easily imagine situations where it is desirable to investigate the co-occurrence frequencies of three modes and beyond. In an information retrieval

*This research was carried out while the author was a PhD student at University of Groningen, The Netherlands.

context, one such situation might be the investigation of *words* × *documents* × *authors*. In a Natural Language Processing (NLP) context, one might want to investigate *words* × *dependency relations* × *bag of word context words*, or *verbs* × *subjects* × *direct objects*.

Note that it is not possible to investigate the three-way co-occurrences in a matrix representation form. It is possible to capture the co-occurrence frequencies of a verb with its subjects and its direct objects, but one cannot capture the co-occurrence frequencies of the verb appearing with the subject and the direct object *at the same time*. When the actual three-way co-occurrence data is 'matricized', valuable information is thrown away. To be able to capture the mutual dependencies among the three modes, we will make use of a generalized *tensor* representation. The two-way co-occurrence models (such as latent semantic analysis) have often been augmented with some form of dimensionality reduction in order to counter noise and overcome data sparseness. We will also make use of dimensionality reduction algorithms appropriate for tensor representations.

In this paper, we will focus on the benefit of three-way methods in the framework of selectional preference induction. Selectional preferences are a useful and versatile resource for a number of applications, such as syntactic disambiguation (Hindle and Rooth 1993), semantic role labeling (Gildea and Jurafsky 2002), and word sense disambiguation (McCarthy and Carroll 2003). The selectional preference of a verb can be defined as the semantic restrictions that the verb imposes on its arguments (and thus the preference it has for particular semantic classes).[1] A verb like *drink*, for example, typically prefers animate subjects and drinkable objects. A selectional preference model keeps track of these semantic classes that verbs prefer for their argument slots.

The standard selectional preference models take the form of a mapping σ: $(v, r, c) \mapsto a$, that maps each selectional tuple (v, r, c) to a real number a, representing the degree of preference of a verb v for a class c with respect to role r (Light and Greiff 2002). Keeping track of single relationships, however, is not enough to build a sufficiently rich model of selectional preferences. Compare the following sentences:

(1) The skyscraper is playing coffee.
(2) The turntable is playing the piano.

The first sentence is a clear violation of the selectional preferences of *play*, both for its subject and object slot: a skyscraper doesn't play, nor is coffee something that can be played. The standard selectional preference models are able to capture this violation, because both *(play, su, skyscraper)* and *(play, obj, coffee)* will not be deemed very plausible by these models. The second sentence, however, is more complicated. The sentence still constitutes a violation of the selectional preferences (turntables are not able to play piano's), but the violation is due to the ambiguity of the verb *play*, and the individual preferences *(play, su, turntable)* and *(play, obj, piano)* are perfectly possible. The standard models, therefore, will not be able to

[1] Note that we will only consider the selectional preference of verbs, although the notion extends to other word classes as well (e.g. the selectional preference of nouns for certain adjectives).

capture this violation. By keeping track of the three-way co-occurrence data, we hope to build a richer model of selectional preferences that does justice to preference violations like the one in sentence 1. Moreover, by using a factorization technique suitable for three-way co-occurrences, we hope to build a three-way selectional preference model that is able to generalize to unseen data.

The paper is organized as follows. In the next section, we give an overview of previous work related to this paper; we discuss previous approaches to selectional preferences and verb clustering, and give an overview of factorization algorithms that have been used for language processing. Section 3 discusses the methodology of three-way factorizations, including an overview of tensors and their algebra. In Section 4, we describe the application of the three-way factorization model for the induction of selectional preferences, and illustrate the kind of data the model is able to capture with some examples. Section 5, then, presents a quantitative evaluation of the model using a pseudo-disambiguation task, and compares the results to other approaches. Section 6, finally, wraps up with some conclusions and pointers for future work.

2 Previous work

2.1 Selectional preferences and verb clustering

Selectional preferences have been a popular research subject in the NLP community. Research on the subject dates back to the early nineties, with early approaches given in Basili, Pazienza and Velardi (1992) and Grishman and Sterling (1992). One of the most influential approaches to the automatic induction of selectional preferences from corpora is the work of Resnik (1993, 1996). Resnik generalizes among nouns by using WordNet noun synsets as semantic classes. He then calculates the *selectional preference strength* $S_{r(v)}$ of a specific verb v in a particular relation r by computing the Kullback–Leibler divergence between the class distribution of the verb $p(c \mid v)$ and the aggregate class distribution $p(c)$:

$$S_{r(v)} = \sum_c p(c \mid v)\ log \frac{p(c \mid v)}{p(c)} \tag{1}$$

The *selectional association* $A_{r(v,c)}$ is then the contribution of a particular semantic class c to the verb's preference strength:

$$A_{r(v,c)} = \frac{p(c \mid v)\ log \frac{p(c|v)}{p(c)}}{S_{r(v)}} \tag{2}$$

The model's generalization relies entirely on WordNet; there is no generalization among the verbs. Other notable approaches using WordNet for selectional preference induction are given by Abe and Li (1996) and Clark and Weir (2001).

The research in this paper is related to previous work on clustering. Pereira, Tishby and Lee (1993) use an information-theoretic-based clustering approach, clustering nouns according to their distribution as direct objects among verbs, conditioned on a set of latent semantic classes

$$p(v, n) = \sum_c p(c, v, n) = \sum_c p(c)p(v \mid c)p(n \mid c) \tag{3}$$

Their model is an asymmetric, one-sided clustering model: only the direct objects are clustered, there is no clustering among the verbs.

Rooth *et al.* (1999) use an Expectation-Maximization (EM)-based clustering technique to induce a clustering based on the co-occurrence frequencies of verbs with their subjects and direct objects. Their clustering model is the same as the one used by Pereira *et al.* (1993), but they have embedded the model in a formal EM-clustering framework. Rooth *et al.*'s (1999) clustering is two-sided: the verbs as well as the subjects/direct objects are clustered. We will use a similar model for evaluation purposes.

Recent approaches using distributional similarity methods for the induction of selectional preferences are the ones given by Bhagat, Pantel and Hovy (2007), Basili *et al.* (2007) and Erk (2007). Erk (2007) uses corpus-based similarity measures for the induction of selectional preferences. The selectional preference S_{r_v} for an argument slot r of a particular verb v and a possible headword w_0 is computed as the weighted sum of similarities between w_0 and the headwords seen as argument fillers for the verb ($Seen(r_v)$); wt_{r_v} is an appropriate weighting function

$$S_{r_v}(w_0) = \sum_{w \in Seen(r_v)} sim(w_0, w) \cdot wt_{r_v}(w) \tag{4}$$

Both *Basili et al.* (2007) and Bhagat *et al.* (2007) investigate the induction of selectional preferences in the context of textual entailment tasks.

This research differs from the above-mentioned approaches by its use of multi-way data: where the above approaches limit themselves to two-way co-occurrences, this research will extend the notion of selectional preferences for multi-way co-occurrences.

2.2 Factorization methods

2.2.1 Two-way factorizations

A number of factorization methods have been employed in natural language processing applications, mainly for the computation of semantic similarity. One of the best-known methods using a two-way factorization is latent semantic analysis (LSA) (Landauer and Dumais 1997; Landauer, Foltz and Laham 1998). Latent semantic analysis models the meaning of words and documents by projecting them into a vector space of reduced dimensionality; the reduced vector space is built up by applying the singular value decomposition (SVD) – a well-known linear algebraic method – to a term-by-document frequency matrix. The resulting lower dimensional matrix is the best possible fit in a least squares sense. By enforcing a lower number of dimensions, the algorithm is forced to make generalizations over the simple frequency data. The Co-occurring terms are mapped to the same dimensions; terms that do not co-occur are mapped to different dimensions. LSA allegedly finds 'latent semantic dimensions', according to which nouns and documents can be represented more efficiently.

While rooted in linear algebra, singular value decomposition has proven to be a useful tool in statistical applications. It is closely akin to statistical methods, such

as principal components analysis, and has been used as a versatile dimensionality reduction technique in different scientific fields, such as image recognition, information retrieval and signal processing (Deprettere 1988). SVD stems from a well-known theorem in linear algebra: A rectangular matrix can be decomposed into three other matrices of specific forms, so that the product of these three matrices is equal to the original matrix:

$$\mathbf{A}_{m\times n} = \mathbf{U}_{m\times z}\ \Sigma_{z\times z}\ (\mathbf{V}_{n\times z})^T \tag{5}$$

where $z = min(m, n)$.

The singular value decomposition can be interpreted as a method that rotates the axes of the n-dimensional space in such a way that the largest variation is captured by the leading dimensions. The diagonal matrix Σ contains the singular values sorted in descending order. Each singular value represents the amount of variance that is captured by a particular dimension. The left-singular and right-singular vectors linked to the highest singular value represent the most important dimension in the data (i.e. the dimension that explains the most variance of the matrix); the singular vectors linked to the second highest value represent the second most important dimension (orthogonal to the first one) and so on. Typically, one uses only the first $k \ll z$ dimensions, stripping off the remaining singular values and singular vectors. If one or more of the least significant singular values are omitted, then the reconstructed matrix will be the best possible least-squares approximation of the original matrix in the lower dimensional space. Intuitively, SVD is able to transform the original matrix – with an abundance of overlapping dimensions – into a new, many times smaller matrix that is able to describe the data in terms of its principal components. Due to this dimension reduction, a more succinct and more general representation of the data is obtained. Redundancy is filtered out, and data sparseness is reduced.

Latent semantic analysis has been criticized for a number of reasons, one of them being the fact that the factorization contains negative numbers. It is not clear what negativity on a semantic scale should designate. Subsequent methods, such as probabilistic latent semantic analysis (Hofmann 1999) and non-negative matrix factorization (Lee and Seung 2000), remedy these problems, and indeed get much more clear-cut semantic dimensions. Below, we discuss the latter method in more detail.

The non-negative matrix factorization (NMF) is a method that factorizes a matrix $\mathbf{V}$ into two other matrices, $\mathbf{W}$ and $\mathbf{H}$:

$$\mathbf{V}_{n\times m} \approx \mathbf{W}_{n\times r}\mathbf{H}_{r\times m} \tag{6}$$

Typically, r is much smaller than n, m so that both instances and features are expressed in terms of a few components. The NMF enforces the constraint that all three matrices must be non-negative, so that all elements must be greater than or equal to zero. The factorization is carried out by minimizing an objective function, typically the least squares error (7):

$$\min\ \|\ \mathbf{V} - \mathbf{WH}\ \|_F = \min\ \sum_i\ \sum_j (\mathbf{V}_{ij} - (\mathbf{WH})_{ij})^2 \tag{7}$$

Practically, the factorization can be efficiently computed through the iterative application of multiplicative update rules. The set of update rules that minimize the least square errors are given in (8) and (9):

$$\mathbf{H}_{a\mu} \leftarrow \mathbf{H}_{a\mu} \frac{(\mathbf{W}^T\mathbf{V})_{a\mu}}{(\mathbf{W}^T\mathbf{W}\mathbf{H})_{a\mu}} \tag{8}$$

$$\mathbf{W}_{ia} \leftarrow \mathbf{W}_{ia} \frac{(\mathbf{V}\mathbf{H}^T)_{ia}}{(\mathbf{W}\mathbf{H}\mathbf{H}^T)_{ia}} \tag{9}$$

Matrices **W** and **H** are randomly initialized, and the update rules are iteratively applied – alternating between them. In each iteration, the matrices **W** and **H** are suitably normalized, so that the rows of the matrices sum to 1. The algorithm stops after a fixed number of iterations, or according to some stopping criterion (the change of the objective function drops below a certain threshold). The update rules are guaranteed to converge to a local optimum.

We will use both singular value decomposition and non-negative matrix factorization for reasons of comparison in our evaluation framework.

2.2.2 Three-way factorizations

To be able to cope with three-way data, several algorithms have been developed as multilinear generalizations of the SVD. In statistics, the three-way component analysis has been extensively investigated (for an overview, see Kiers and van Mechelen 2001). The two most popular methods are the parallel factor analysis (PARAFAC) (Carroll and Chang 1970; Harshman 1970) and the Tucker decomposition (Tucker 1966). Three-way factorizations have been applied in various domains, such as psychometry and image recognition (Vasilescu and Terzopoulos 2002). In information retrieval, three-way factorizations have been applied to the problem of link analysis (Kolda and Bader 2006).

One last important method dealing with multi-way data is the non-negative tensor factorization (NTF) (Shashua and Hazan 2005). NTF is a generalization of non-negative matrix factorization, and can be considered an extension of the PARAFAC model with the constraint of non-negativity (cfr. infra).

One of the few papers that has investigated the application of tensor factorization for NLP is Turney (2007), in which a three-mode tensor is used to compute the semantic similarity of words. The method achieves 83.75 percent accuracy on the TOEFL synonym questions.

3 Methodology

3.1 Three-way data

The distributional similarity methods usually treat language data as two-way occurrences, represented in the form of a *matrix*. The matrix representation is perfectly suited for the representation of two-way co-occurrence phenomena; it allows for the application of algebraic and statistical methods, which in turn allow for

$$\begin{array}{llll} \langle & \text{speel} & \text{su} & \text{voetbalclub} \rangle \\ \langle & \text{speel} & \text{obj} & \text{wedstrijd} \rangle \\ \langle & \text{speel} & \text{su} & \text{acteur} \rangle \\ \langle & \text{speel} & \text{obj} & \text{Hamlet} \rangle \end{array}$$

Fig. 1. Extracted dependency relations.

$$\mathrm{V} = \begin{bmatrix} & \textit{voetbalclub}_{su} & \textit{acteur}_{su} & \textit{wedstrijd}_{obj} & \textit{Hamlet}_{obj} & \dots \\ \text{speel} & 1 & 1 & 1 & 1 & \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

Fig. 2. Two-way co-occurrence matrix.

the induction of semantic generalizations. The reduction of language to two-way co-occurrence frequencies, however, is a vast oversimplification. Language is a complex system of interrelated words, driven a.o. by grammar rules and subcategorization frames. By keeping track of multi-way co-occurrences, we can attempt to do some more justice to this complexity (although the framework represented here remains a major simplification). Compare the following sentences:

(3) De voetbalclub$_{su}$ speelt een goede wedstrijd$_{obj}$.
The soccer team plays a good game.
The soccer team is playing a good game.

(4) De acteur$_{su}$ speelt Hamlet$_{obj}$.
The actor plays Hamlet.
The actor is playing Hamlet.

The dependency relations (subject–verb and verb–object) for examples 3.1 and 3.1 are given in Figure 1.

Say, we now want to investigate the semantics of Dutch verbs (like *spelen* 'to play'). In our standard two-way framework, we would then take the verbs to be one mode, and combine the syntactic dependencies that the verbs appear with (subjects and direct objects in this case) together in the other mode, yielding a matrix like the one in Figure 2.

This is a genuine and appropriate way of investigating a verb's semantics, but we do lose some more complex and interesting relations between a verb and its objects. By capturing a verb's co-occurrences in a matrix form, we are able to investigate its co-occurrence with particular subjects and direct objects separately, but we are not able to investigate a verb's co-occurrence with subjects and direct objects *at the same time*: we lose the three-way relationship that exists between verb, subject and direct object.

It is possible, however, to capture in a matrix the relationship between subjects and direct objects for one particular verb. For the above examples with *spelen*, this

$$\mathrm{V}_{\mathrm{speel}} = \begin{bmatrix} & \text{wedstrijd} & \text{Hamlet} & \dots \\ \text{voetbalclub} & 1 & 0 & \dots \\ \text{acteur} & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

Fig. 3. Two-way co-occurrence matrix of subjects and direct objects for the verb *spelen* 'to play'.

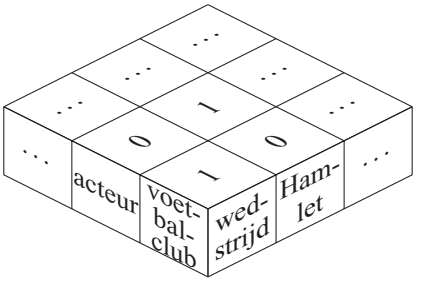

Fig. 4. Graphical representation of the co-occurrence matrix for the verb *spelen* 'to play'.

$$\mathrm{V}_{\mathrm{win}} = \begin{bmatrix} & \text{wedstrijd} & \text{Hamlet} & \dots \\ \text{voetbalclub} & 1 & 0 & \dots \\ \text{acteur} & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

Fig. 5. Two-way co-occurrence matrix of subjects and direct objects for the verb *winnen* 'to win'.

$$\mathrm{V}_{\mathrm{bewerk}} = \begin{bmatrix} & \text{wedstrijd} & \text{Hamlet} & \dots \\ \text{voetbalclub} & 0 & 0 & \dots \\ \text{acteur} & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

Fig. 6. Two-way co-occurrence matrix of subjects and direct objects for the verb *bewerken* 'to adapt'.

yields the matrix given in Figure 3. The rows of the matrix represent the subjects, and the columns represent the direct objects. A graphical representation of this matrix is given in Figure 4.

We can now construct similar matrices for each verb that we want to investigate. Say, we want two include two more verbs, *winnen* 'to win', and *bewerken* 'to adapt'. The matrices for these verbs are given in Figures 5 and 6. Note that the two matrices $\mathbf{V}_{\mathrm{win}}$ and $\mathbf{V}_{\mathrm{bewerk}}$ contain the same instances and features (subjects and direct objects) in their rows and columns as the first matrix $\mathbf{V}_{\mathrm{speel}}$.

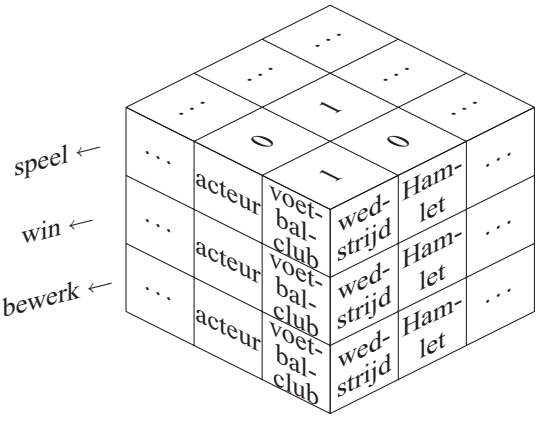

Fig. 7. Graphical representation tensor.

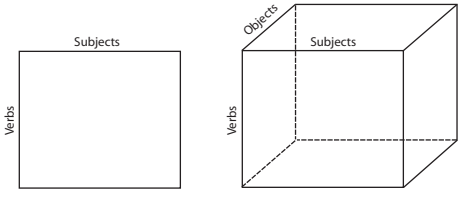

Fig. 8. Matrix representation versus tensor representation.

As a last step, we can now stack the three matrices together, which yields a three-dimensional cube like the one shown in Figure 7.

It would be clear by now that we are leaving the familiar domain of two-dimensional matrices. As we have noted before, a matrix is not a suitable form for the representation of multi-way data. For co-occurrence data beyond two modes, we need a more general representation. The generalization of a matrix is called a *tensor*. A tensor is able to encode co-occurrence data of any n modes. Figure 8 shows a graphical comparison of a matrix and a tensor with three modes – although a tensor can easily be generalized to more than three modes.[2]

We are now leaving the familiar domain of two-dimensional matrix algebra, and enter the realm of higher dimensional tensor algebra. Tensor algebra involves some novel mathematical machinery. The next section provides a succinct introduction to tensor algebra.

3.2 Tensor algebra

In this overview of tensor algebra, we will review some conceptual and notational preliminaries based on Kiers (2000) and Kolda and Bader (2009), and focus on some

[2] We will stick to examples that use no more than three modes; it is easy to construct higher order tensors mathematically, but it is not possible to represent them visually.

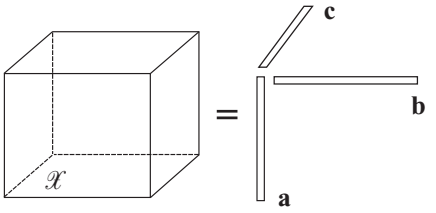

Fig. 9. A third-order rank one tensor, which can be written as the outer product of three vectors.

vital tensor operations required for the tensor-based factorizations explained in the next section.

The *order* of a tensor is its number of 'dimensions'.[3] Vectors (tensors of order one) are denoted by boldface lowercase letters (**x**). Matrices (tensors of order two) are denoted by boldface capital letters (**X**). Higher order tensors (order three or higher) are denoted by boldface calligraphic letters ($\mathcal{X}$). Scalars are denoted by lowercase letters (x).

The ith entry of a vector **x** is written as x_i, element (i, j) of a matrix **X** is written as x_{ij} and element (i, j, k) of a third-order tensor $\mathcal{X}$ is written as x_{ijk}. Indices range from 1 to D, so that $i = 1, \ldots, D$. The nth element in a sequence is written as a superscript in parentheses, so $\mathbf{A}^{(n)}$ denotes the nth matrix in a sequence.

The *norm* of a tensor $\mathcal{X} \in \mathbb{R}^{D_1 \times D_2 \times D_3}$ – analogous to the Frobenius norm for matrices – is the square root of the sum of squares of all its elements (10):

$$\| \mathcal{X} \| = \sqrt{\sum_{i_1=1}^{D_1} \sum_{i_2=1}^{D_2} \cdots \sum_{i_N=1}^{D_N} x_{i_1 i_2 \ldots i_N}^2} \tag{10}$$

An N-order tensor $\mathcal{X} \in \mathbb{R}^{D_1 \times D_2 \times \ldots \times D_N}$ is of *rank one* if it can be written as the outer product of N vectors:

$$\mathcal{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \ldots \circ \mathbf{a}^{(N)} \tag{11}$$

The small circle ($\circ$) denotes the outer product and is calculated by multiplying for each element of the tensor the corresponding vector elements:

$$x_{i_1 i_2 \ldots i_N} = a_{i_1}^{(1)} a_{i_2}^{(2)} \ldots a_{i_N}^{(N)} \quad \text{for all } 1 \leq i_n \leq I_n \tag{12}$$

A third-order rank one tensor is given in Figure 9.

Finally, the *rank* of a tensor $\mathcal{X}$ is defined as the smallest number of rank one tensors whose sum is equal to $\mathcal{X}$. Figure 10 shows a tensor that can be generated with the sum of three rank one tensors (the sum of three outer products). Therefore, $\text{rank}(\mathcal{X}) = 3$.

[3] The term *dimension* is ambiguous, because it is also used to denote the cardinality of vectors and vector spaces. Therefore, the dimensions of a tensor are usually referred to as *modes* or *ways*.

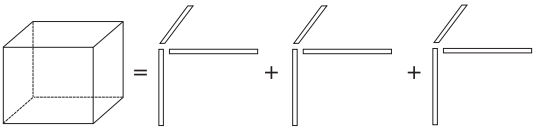

Fig. 10. A third-order tensor of rank three, which can be written as the sum of three outer products.

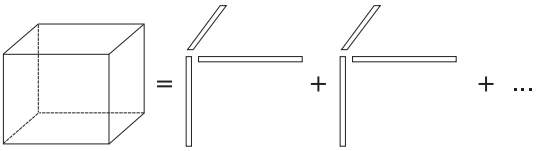

Fig. 11. Graphical representation of PARAFAC as the sum of outer products.

3.3 Multi-way factorization algorithms

In the next sections, we will look in some more detail at two different (but related) multi-way factorization algorithms: parallel factor analysis and non-negative tensor factorization. An extensive overview of multi-way data analysis is given in Acar and Yener (2009) and Kolda and Bader (2009).

3.3.1 Parallel factor analysis

The parallel factor analysis (PARAFAC) is a multilinear analogue of SVD used in latent semantic analysis. The key idea is to minimize the sum of squares between the original tensor and the factorized model of the tensor. For the three mode case of a tensor $\mathcal{T} \in \mathbb{R}^{D_1 \times D_2 \times D_3}$ this gives the objective function in 13, where k is the number of dimensions in the factorized model (recall that $\circ$ denotes the outer product):

$$\min_{x_i \in \mathbb{R}^{D1}, y_i \in \mathbb{R}^{D2}, z_i \in \mathbb{R}^{D3}} \left\| \mathcal{T} - \sum_{i=1}^{k} x_i \circ y_i \circ z_i \right\| \tag{13}$$

The factorization algorithm finds a tensor (as a sum of rank one tensors) that is as similar as possible to the original tensor in the least squares sense, but has a fixed lower rank k: it is the best possible low-rank approximation of the original tensor for rank k.

The algorithm results in three matrices, indicating the loadings of each mode on the factorized dimensions. The model is graphically represented in Figure 11, visualizing the fact that the PARAFAC decomposition consists of the summation over the outer products of n (in this case three) vectors.

Figure 12 represents the factorization as three loading matrices, containing the loadings on each factor for the three different modes. The representation is equivalent to the representation with the sum of outer products – it differs only conceptually.

There are a number of algorithms available to calculate the PARAFAC decomposition. The most popular one is the alternating least squares method (ALS), which was proposed in the original papers by Carroll and Chang (1970) and Harshman (1970). In each iteration, two of the modes are fixed and the third one is fitted in a least

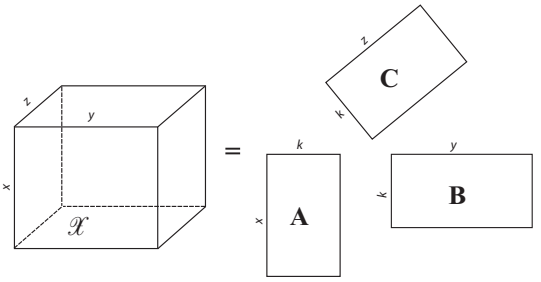

Fig. 12. Graphical representation of PARAFAC as three loading matrices.

squares sense. This calculation is done for each mode in turn, and the process is repeated until convergence.

3.3.2 Non-negative tensor factorization

Our second multi-way factorization is called the non-negative tensor factorization (NTF), it is the generalization of the non-negative matrix factorization for multi-way data. The NTF model is similar to the PARAFAC analysis, with the constraint that all data needs to be non-negative (i.e. ≥ 0). This yields the objective function in (14):

$$\min_{x_i \in \mathbb{R}^{D1}_{\geq 0}, y_i \in \mathbb{R}^{D2}_{\geq 0}, z_i \in \mathbb{R}^{D3}_{\geq 0}} \left\| \mathscr{T} - \sum_{i=1}^{k} x_i \circ y_i \circ z_i \right\| \tag{14}$$

As with the PARAFAC model, the algorithm results in three matrices, indicating the loadings of each mode on the factorized dimensions.

There are again a number of ways to compute the factorization. Bro and Jong (1997) use an alternating least squares algorithm similar to the ALS algorithm for PARAFAC, but with some adaptations to enforce non-negativity. The non-negativity can be enforced by using a non-negative least squares computation, based on Lawson and Hanson (1974).[4] Another possibility is to use multiplicative update rules – similar to the update rules for non-negative matrix factorization, explained in Section 2.2.1 (Welling and Weber 2001).

4 Application

4.1 Three-way selectional preferences

The model can straightforwardly be applied to language data. In this part, we describe the factorization of *verbs* × *subjects* × *direct objects* co-occurrences, but the example can easily be substituted with other co-occurrence information. Moreover, the model need not be restricted to three modes; it is very well possible to go to four modes and beyond – as long as the computations remain feasible.

[4] The algorithm used in this research is a non-negative ALS algorithm that has been implemented in MATLAB, using the Tensor Toolbox for sparse tensor calculations (Bader and Kolda 2009).

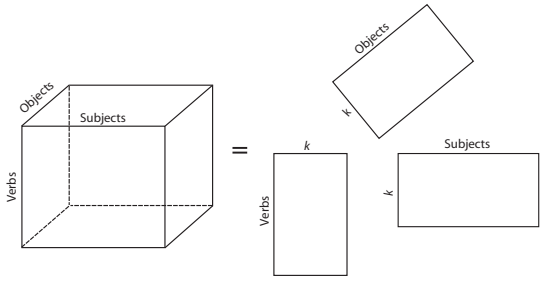

Fig. 13. Graphical representation of NTF for language data.

The NTF decomposition for the *verbs* × *subjects* × *direct objects* co-occurrences into the three loading matrices is graphically represented in Figure 13. By applying the NTF model to three-way (s, v, o) co-occurrences, we want to extract a generalized selectional preference model, and eventually even induce some kind of frame semantics (in the broad sense of the word).

In the resulting factorization, each verb, subject and direct object get a loading value for each factor dimension in the corresponding loadings matrix. The original value for a particular (s, v, o) triple x_{svo} can then be reconstructed with (15):

$$x_{svo} = \sum_{i=1}^{k} s_{si} v_{vi} o_{oi} \tag{15}$$

To reconstruct the selectional preference value for the triple $(man, bite, dog)$, for example, we look up the subject vector for *man*, the verb vector for *bite* and the direct object vector for *dog*. Then, for each dimension i in the model, we multiply the ith value of the three vectors. The sum of these values is the final preference value.

4.2 Methodological remarks

The approach described in the previous section has been applied to Dutch, using the Twente Nieuws Corpus (Ordelman 2002), a 500-million words corpus of Dutch newspaper texts. The corpus has been parsed with the Dutch-dependency parser Alpino (van Noord 2006),[5] and three-way co-occurrences of verbs with their respective subject and direct object relations have been extracted. As dimension sizes, the 1kmost frequent verbs were used, together with the 10k most frequent subjects and 10k most frequent direct objects, yielding a tensor of $1k \times 10k \times 10k$. The resulting tensor is very sparse, with only .02 percent (1/5,000 th) of the values being non-zero.

The tensor has been adapted with a straightforward extension of pointwise mutual information (Church and Hanks 1990) for three-way co-occurrences, following (16).

[5] The Alpino parser is able to parse newspaper texts with an accuracy above 90 percent.

Table 1. *Top 10 subjects, verbs and direct objects for the 'police action' dimension*

Subjects	su_s	Verbs	v_s	Objects	obj_s
politie 'police'	.99	*houd aan* 'arrest'	.64	*verdachte* 'suspect'	.16
agent 'policeman'	.07	*arresteer* 'arrest'	.63	*man* 'man'	.16
autoriteit 'authority'	.05	*pak op* 'run in'	.41	*betoger* 'demonstrator'	.14
Justitie 'Justice'	.05	*schiet dood* 'shoot'	.08	*relschopper* 'rioter'	.13
recherche 'detective force'	.04	*verdenk* 'suspect'	.07	*raddraaiers* 'instigator'	.13
marechaussee military police'	.04	*tref aan* 'find'	.06	*overvaller* 'raider'	.13
justitie 'justice'	.04	*achterhaal* 'overtake'	.05	*Roemeen* 'Romanian'	.13
arrestatieteam 'special squad'	.03	*verwijder* 'remove'	.05	*actievoerder* 'campaigner'	.13
leger 'army'	.03	*zoek* 'search'	.04	*hooligan* 'hooligan'	.13
douane 'customs'	.02	*spoor op* 'track'	.03	*Algerijn* 'Algerian'	.13

Negative values are set to zero.[6]

$$MI3(x, y, z) = \log \frac{p(x, y, z)}{p(x)p(y)p(z)} \tag{16}$$

The resulting matrix has been factorized into k dimensions (varying between 50 and 300) with the NTF algorithm described in Section 3.3.2.

4.3 Examples

Tables 1 to 5 show example dimensions that have been found by the algorithm with $k = 100$. Each example gives the top 10 subjects, verbs and direct objects for a particular dimension, together with the score for that particular dimension. These example dimensions are given to illustrate the model's ability to extract an initial form of frame semantics.

Table 1 shows the induction of a 'police action' frame, with police authorities as subjects, police actions as verbs and patients of the police actions as direct objects.

In Table 2, a legislation dimension is induced, with legislative bodies as subjects,[7] legislative actions as verbs and mostly law (proposals) as direct objects. Note that some direct objects (e.g. *minister*) also designate persons that can be the object of a legislative act.

Table 3 depicts a dimension of 'war deployment'. The dimension contains military powers (countries, military organizations and leaders) that deploy (or remove) some form of military force.

[6] This is not just an ad hoc conversion to enforce non-negativity. Negative values indicate a smaller co-occurrence probability than the expected number of co-occurrences. Setting those values to zero proves beneficial for similarity calculations (see, e.g. Bullinaria and Levy 2007).

[7] Note that VVD, D66, PvdA and CDA are Dutch political parties.

Table 2. *Top 10 subjects, verbs and direct objects for the 'legislation' dimension*

Subjects	su_s	Verbs	v_s	Objects	obj_s
meerderheid 'majority'	.33	*steun* 'support'	.83	*motie* 'motion'	.63
VVD	.28	*dien in* 'submit'	.44	*voorstel* 'proposal'	.53
D66	.25	*neem aan* 'pass'	.23	*plan* 'plan'	.28
Kamermeerderheid 'Chamber majority'	.25	*wijs af* 'reject'	.17	*wetsvoorstel* 'bill'	.19
fractie 'party'	.24	*verwerp* 'reject'	.14	*hem* 'him'	.18
PvdA	.23	*vind* 'think'	.08	*kabinet* 'cabinet'	.16
CDA	.23	*aanvaard* 'accepts'	.05	*minister* 'minister'	.16
Tweede Kamer 'Second Chamber'	.21	*behandel* 'treat'	.05	*beleid* 'policy'	.13
partij 'party'	.20	*doe* 'do'	.04	*kandidatuur* 'candidature'	.11
Kamer 'Chamber'	.20	*keur goed* 'pass'	.03	*amendement* 'amendment'	.09

Table 3. *Top 10 subjects, verbs and direct objects for the 'war movement' dimension*

Subjects	su_s	Verbs	v_s	Objects	obj_s
regering 'government'	.26	*stuur* 'send'	.72	*troep* 'troop'	.82
VS 'US'	.26	*trek terug* 'withdraw'	.67	*militair* 'soldier'	.42
Nederland 'Netherlands'	.25	*zet in* 'deploy'	.14	*soldaat* 'soldier'	.21
president 'president'	.23	*lever* 'supply'	.07	*delegatie* 'delegation'	.12
leger 'army'	.22	*heb* 'have'	.06	*leger* 'army'	.12
land 'country'	.20	*haal weg* 'remove'	.03	*waarnemer* 'observer'	.08
NAVO 'NATO'	.20	*beschikbaar stel* 'make available'	.03	*marinier* 'marine'	.08
Indonesië 'Indonesia'	.19	*zeg toe* 'promise'	.03	*grondtroepen* 'ground forces'	.08
Verenigde Staten 'United States'	.19	*ruim op* 'clear'	.02	*gezant* 'envoy'	.07
Groot-Brittannië 'Great Britain'	.18	*bied aan* 'offer'	.02	*versterking* 'reinforcement'	.07

Table 4 shows a 'publishing' dimension. The subjects contain writers (persons and bodies), who publish (in a broad sense) textual works.

Table 5, finally, is clearly an exhibition dimension, with verbs describing actions of display and trade that art institutions (subjects) can perform with works of art (objects).

These are not the only sensible dimensions that have been found by the algorithm. A quick qualitative evaluation indicates that about 44 dimensions contain similar, framelike semantics. In another 43 dimensions, the semantics are less clear-cut (single verbs or expressions account for one dimension, or different senses of a verb get mixed up). Thirteen dimensions are not very much based on semantic characteristics, but rather on syntax (e.g. fixed expressions and pronomina).

Table 4. *Top 10 subjects, verbs and direct objects for the 'publishing' dimension*

Subjects	su_s	Verbs	v_s	Objects	obj_s
hij 'he'	.62	*schrijf* 'write'	.87	*boek* 'book'	.30
die 'who'	.41	*publiceer* 'publish'	.33	*roman* 'novel'	.21
ze 'she'	.32	*zing* 'sing'	.14	*brief* 'letter'	.20
ik 'I'	.30	*lees voor* 'read aloud'	.09	*gedicht* 'poem'	.18
zij 'she'	.19	*wijd* 'devote'	.09	*tekst* 'text'	.17
auteur 'author'	.16	*vertaal* 'translate'	.09	*essay* 'essay'	.17
je 'you'	.14	*bewerk* 'adapt'	.08	*stuk* 'piece'	.16
journalist 'journalist'	.13	*voltooi* 'finish'	.07	*artikel* 'article'	.16
schrijver 'writer'	.13	*componeer* 'compose'	.06	*biografie* 'biography'	.15
krant 'newspaper'	.09	*presenteer* 'present'	.06	*verhaal* 'story'	.14

Table 5. *Top 10 subjects, verbs and direct objects for the 'exhibition' dimension*

Subjects	su_s	Verbs	v_s	Objects	obj_s
tentoonstelling 'exhibition'	.50	*toon* 'display'	.72	*schilderij* 'painting'	.47
expositie 'exposition'	.49	*omvat* 'cover'	.63	*werk* 'work'	.46
galerie 'gallery'	.36	*bevat* 'contain'	.18	*tekening* 'drawing'	.36
collectie 'collection'	.29	*presenteer* 'present'	.17	*foto* 'picture'	.33
museum 'museum'	.27	*laat* 'let'	.07	*sculptuur* 'sculpture'	.25
oeuvre 'oeuvre'	.22	*koop* 'buy'	.07	*aquarel* 'aquarelle'	.20
Kunsthal	.19	*bezit* 'own'	.06	*object* 'object'	.19
kunstenaar 'artist'	.15	*zie* 'see'	.05	*beeld* 'statue'	.12
dat 'that'	.12	*koop aan* 'acquire'	.05	*overzicht* 'overview'	.12
hij 'he'	.10	*in huis heb* 'own'	.04	*portret* 'portrait'	.11

The qualitative evaluation indicates that some dimensions indeed do not contain framelike semantics, but those do contain information that may be useful for selectional preference induction. Such a dimension is shown in Table 6. It shows an example dimension in which practically all of the dimension's mass is attributed to one particular expression: *een rol spelen*, 'to play a role'. The subject slot is more spread out: different kind of things might play a role – each with a fairly low probability.

5 Evaluation

5.1 Evaluation framework

The results of the NTF model have been quantitatively evaluated in a pseudo-disambiguation task, similar to the one used by Rooth *et al.* (1999). It is used to evaluate the generalization capabilities of the algorithm. The task is to judge which subject (s or s') and direct object (o or o') is more likely for a particular verb v, where (s, v, o) is a combination drawn from the corpus, and s' and o' are a

Table 6. *Top 10 subjects, verbs and direct objects for the 'play a role' dimension*

Subjects	su_s	Verbs	v_s	Objects	obj_s
naamsbekendheid 'fame'	.04	*speel* 'play'	1.00	*rol* 'role'	1.00
aard 'nature'	.04	*lever op* 'yield'	.00	*hoofdrol* 'leading part'	.03
nut 'use'	.04	*onderzoek* 'research'	.00	*die* 'who'	.00
hygiëne 'hygiene'	.04	*zie* 'see'	.00	*stroming* 'movement'	.00
eerwraak 'revenge'	.04	*neem* 'take'	.00	*hoofd* 'head'	.00
schaamte 'shame'	.04	*vertolk* 'express'	.00	*religie* 'religion'	.00
institutie 'institution'	.04	*dring terug* 'push back'	.00	*werk* 'work'	.00
Cultuur 'Culture'	.04	*hekel* 'criticize'	.00	*traditie* 'tradition'	.00
verdeling 'division'	.04	*krijg* 'get'	.00	*overheid* 'government'	.00
verbinding 'connection'	.04	*onderstreep* 'underline'	.00	*iedereen* 'everyone'	.00

Table 7. *Three examples from the pseudo-disambiguation evaluation task's test set*

s	v	o	s'	o'
jongere 'youngster'	*drink* 'drink'	*bier* 'beer'	*coalitie* 'coalition'	*aandeel* 'share'
werkgever 'employer'	*riskeer* 'risk'	*boete* 'fine'	*doel* 'goal'	*kopzorg* 'worry'
directeur 'manager'	*zwaai* 'sway'	*scepter* 'sceptre'	*informateur* 'informer'	*vodka* 'wodka'

subject and direct object randomly drawn from the corpus. A triple is considered correct if the algorithm prefers both s and o over their counterparts s' and o' (so the (s, v, o) triple – that appears in the test corpus – is preferred over the triples (s', v, o'), (s', v, o) and (s, v, o'), according to (15)). Table 7 shows three examples from the pseudo-disambiguation task.

Four different models have been evaluated. The first two models are tensor factorization models. The first model is the NTF model, as described in Section 3.3.2. The second model is the original PARAFAC model, without the non-negativity constraints.

The other two models are matrix factorization models. The third model is the NMF model, and the fourth model is the SVD model. For these models, a matrix has been constructed that contains the pairwise co-occurrence frequencies of verbs by subjects as well as direct objects. This gives a matrix of 1k verbs by 10k subjects + 10k direct objects (1k $\times$ 20k). The matrix has been adapted with pointwise mutual information.

The models have been evaluated with 10-fold cross-validation. The corpus contains 298,540 different (s, v, o) co-occurrences. Those have been randomly divided into ten equal parts. So in each fold, 268,686 co-occurrences have been used for training, and 29,854 for testing.

Table 8. *Results of the 10-fold cross-validation for the* NTF, PARAFAC, NMF *and* SVD *model for 50, 100 and 300 factors (averages and standard deviation)*

Algorithm		Accuracy (%) ± standard deviation		
		50 factors	100 factors	300 factors
3-way	NTF	**89.52** ± 0.18	**90.43** ± 0.14	**90.89** ± 0.16
	PARAFAC	85.57 ± 0.25	83.58 ± 0.59	80.12 ± 0.76
2-way	NMF	81.79 ± 0.15	78.83 ± 0.40	75.74 ± 0.63
	SVD	69.60 ± 0.41	62.84 ± 1.30	45.22 ± 1.01

5.2 Evaluation results

The accuracy results of the evaluation are given in Table 8. The results clearly indicate that the NTF model outperforms all the other models. The model achieves the best result with 300 dimensions, but the differences between the different NTF models are not very large – all attaining scores around 90 percent.

The PARAFAC results indicate the fitness of tensor factorization for the induction of three-way selectional preferences. Even without the constraint of non-negativity, the model outperforms the matrix factorization models, reaching a score of about 85 percent. The model deteriorates when more dimensions are used.

Both matrix factorization models perform worse than their tensor factorization counterparts. The NMF model still scores reasonably well, indicating the positive effect of the non-negativity constraint. The simple SVD model performs worst, reaching a score of about 70 percent with fifty dimensions.

6 Conclusion and future work

This paper has presented a novel method that is able to investigate three-way co-occurrences. Other distributional methods deal almost exclusively with pairwise co-occurrences. The ability to keep track of multi-way co-occurrences opens up new possibilities and brings about interesting results. The three-way factorization methods are able to generalize among the data and overcome data sparseness.

The method has been applied to the problem of selectional preference induction. The results indicate that the algorithm is able to induce selectional preferences, leading to a broad kind of frame semantics. The quantitative evaluation shows that the use of three-way data is clearly beneficial for the induction of three-way selectional preferences. The tensor models outperform the simple matrix models in the pseudo-disambiguation task. The results also indicate the positive effect of the non-negativity constraint: both models with non-negative constraints outperform their non-constrained counterparts.

The results and the evaluation indicate that the method presented here is a promising tool for the investigation of NLP topics, although more research and thorough evaluation are desirable.

There is quite some room for future work. First of all, we want to further investigate the usefulness of the method for selectional preference induction. One of the most important extensions is the inclusion of other dependency relations in our model, apart from subjects and direct objects (thus making use of tensors with more than three modes).

Secondly, there is room for improvement and further research with regard to the tensor factorization model. The model presented here minimizes the sum of squared distance. This is, however, not the only objective function possible. Another possibility is the minimization of the Kullback–Leibler divergence. Minimizing the sum of squared distance normally assumes distributed data, and language phenomena are rarely normally distributed. Other objective functions – such as the minimization of the Kullback–Leibler divergence – might be able to capture the language structures much more adequately. We specifically want to stress this second line of future research as one of the most promising and exciting ones.

Finally, the model presented here is not only suitable for selectional preference induction. There are many problems in NLP that involve three-way co-occurrences. In future work, we want to apply the NTF model presented here to other problems in NLP, the most important one being word sense discrimination.

Acknowledgments

Brett Bader kindly provided his implementation of non-negative tensor factorization for sparse matrices, from which this research has substantially benefited. Anonymous reviewers provided fruitful comments and remarks on this paper, which considerably improved its quality.

References

Abe, N. and Li, H. 1996. Learning word association norms using tree cut pair models. In *Proceedings of the Thirteenth International Conference on Machine Learning*, Bari, Italy, pp. 3–11.

Acar, E. and Yener, B. 2009. Unsupervised multiway data analysis: A literature survey. *IEEE Transactions on Knowledge and Data Engineering* **21**(1): 6–20.

Bader, B. W. and Kolda, T. G. 2006a. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software* **32**(4), December.

Bader, B. W. and Kolda, T. G. 2006b. Efficient MATLAB computations with sparse and factored tensors. Technical Report SAND2006-7592, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, December.

Bader, B. W. and Kolda, T. G. 2009. Matlab tensor toolbox version 2.3. `http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/`, July.

Basili, R., Pazienza, M. T., and Velardi, P. 1992. Computational lexicons: the neat examples and the odd exemplars. In *Proceedings of Applied Natural Language Processing Conference - ANLP*, Trento, Italy, pp. 96–103.

Basili, R., De Cao, D., Marocco, P., and Pennacchiotti, M. 2007. Learning selectional preferences for entailment or paraphrasing rules. In *Proceedings of RANLP 2007*, Borovets, Bulgaria.

Bhagat, R., Pantel, P., and Hovy, E. 2007. Ledir: an unsupervised algorithm for learning directionality of inference rules. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-07)*, pp. 161–170, Prague, Czech Republic.

Bro, R. and De Jong, S.. 1997. A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics* **11**: 393–401.

Bullinaria, J. A. and Levy, J. P. 2007. Extracting semantic representations from word co-occurrence statistics: a computational study. *Behavior Research Methods* **39**: 510–526.

Carroll, J. D. and Chang, J.-J. 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika* **35**: 283–319.

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

Clark, S. and Weir, D. 2001. Class-based probability estimation using a semantic hierarchy. In *Proceedings of NAACL 2001*, Pittsburgh, USA, pp. 95–102.

Deprettere, F. (ed.) 1988. *SVD and Signal Processing: Algorithms, Applications and Architectures*. Amsterdam, The Netherlands: North-Holland Publishing.

Erk, K. 2007. A simple, similarity-based model for selectional preferences. In *Proceedings of ACL 2007*, Prague, Czech Republic, pp. 216–223.

Gildea, D. and Jurafsky, D. 2002. Automatic labeling of semantic roles. *Computational Linguistics* **28**(3): 245–288.

Grishman, R. and Sterling, J. 1992. Acquisition of selectional patterns. In *Proceedings of COLING 1992*, Nantes, France, pp. 658–664.

Harshman, R. A. 1970. Foundations of the parafac procedure: models and conditions for an "explanatory" multi-mode factor analysis. In *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, Los Angeles: University of California.

Hindle, D. and Rooth, M. 1993. Structural ambiguity and lexical relations. *Computational Linguistics* **19**(1): 103–120.

Hofmann, T. 1999. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, Sweden, pp. 289–296.

Kiers, H. A. L. and van Mechelen, I. 2001. Three-way component analysis: Principles and illustrative application. *Psychological Methods* **6**: 84–110.

Kiers, H. A. L. 2000. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics* **14**: 105–122.

Kolda, T. and Bader, B. 2006. The TOPHITS model for higher-order web link analysis. In *Workshop on Link Analysis, Counterterrorism and Security*, Bethesda, MD, USA.

Kolda, T. G. and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM Review* **51**(3), September.

Landauer, T. and Dumais, S. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychology Review* **104**: 211–240.

Landauer, T., Foltz, P., and Laham, D. 1998. An Introduction to Latent Semantic Analysis. *Discourse Processes* **25**: 295–284.

Lawson, C. L. and Hanson, B. J. 1974. *Solving Least Squares Problems*. Englewood Cliffs, NJ: Prentice-Hall.

Lee, D. D. and Seung, H. S. 2000. Algorithms for non-negative matrix factorization. In *Proceedings of the 2000 Conference of the Advances in Neural information Processing Systems 13*, Denver, CO, USA, pp. 556–562.

Light, M. and Greiff, W. 2002. Statistical models for the induction and use of selectional preferences. *Cognitive Science* **26**: 269–281.

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

Ordelman, R. J. F. 2002. Twente Nieuws Corpus (TwNC), August. Parlevink Language Technology Group, University of Twente, The Netherlands.

Pereira, F., Tishby, N., and Lee, L. 1993. Distributional clustering of English words. In *31st Annual Meeting of the ACL*, Columbus, OH, USA, pp. 183–190.

Resnik, P. S. 1993. *Selection And Information: A Class-based Approach to Lexical Relationships*. Ph.D. thesis, University of Pennsylvania.

Resnik, P. 1996. Selectional constraints: an information-theoretic model and its computational realization. *Cognition* **61**: 127–159, November.

Rooth, M., Riezler, S., Prescher, D., Carroll, G., and Beil, F. 1999. Inducing a semantically annotated lexicon via em-based clustering. In *37th Annual Meeting of the ACL*, College Park, Maryland, USA, pp. 104–111.

Shashua, A. and Hazan, T. 2005. Non-negative tensor factorization with applications to statistics and computer vision. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pp. 792–799, New York, NY, USA: ACM.

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

Turney, P. D. 2007. Empirical evaluation of four tensor decomposition algorithms. Technical Report ERB-1152, Ottawa, ON, Canada: National Research Council, Institute for Information Technology.

van Noord, G. 2006. At Last Parsing Is Now Operational. In Piet Mertens, Cedrick Fairon, Anne Dister, and Patrick Watrin (eds.), *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, pp. 20–42, Leuven, Belgium, Leuven University Press.

Vasilescu, M. A. O. and Terzopoulos, D. 2002. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision (ECCV '02)*, Copenhagen, Denmark, pp. 447–460.

Welling, M. and Weber, M. 2001. Positive tensor factorization. *Pattern Recognition Letters* **22**: 1255–1261.