Exploring Three Way Contexts for Word Sense Discrimination

Tim Van de Cruys

University of Groningen
t.van.de.cruys@rug.nl

Abstract. In this paper, an extension of a dimensionality reduction algorithm called NON-NEGATIVE MATRIX FACTORIZATION is presented that combines 'bag of words' data and syntactic data, in order to find latent semantic dimensions according to which both words and syntactic relations can be classified. The use of three way data allows one to determine which dimension(s) are responsible for a certain sense of a word, and adapt the corresponding feature vector accordingly, 'subtracting' one sense to discover another one. The intuition in this is that the syntactic features of the syntax-based approach can be disambiguated by the latent semantic dimensions found by the bag of words approach.

1 Introduction

Automatically acquiring semantics from text is a subject that has gathered a lot of attention for quite some time now. As Manning and Schütze [1] point out, most work on acquiring semantic properties of words has focused on *semantic similarity.* 'Automatically acquiring a relative measure of how similar a word is to known words [...] is much easier than determining what the actual meaning is.' [1, §8.5]

Most work on semantic similarity relies on the distributional hypothesis [2]. This hypothesis states that words that occur in similar contexts tend to be similar. With regard to the context used, two basic approaches exist. One approach makes use of 'bag of words' co-occurrence data; in this approach, a certain window around a word is used for gathering co-occurrence information. The window may either be a fixed number of words, or the paragraph or document that a word appears in. The window-based method is often augmented with some form of dimensionality reduction, that is able to capture 'latent semantic dimensions' in the data.

The second approach uses a more fine grained distributional model, focusing on the syntactic relations that words appear with. Typically, a large text corpus is parsed, and dependency triples are extracted.[1] Words are considered similar if they appear with similar syntactic relations. Note that the former approach does not need any kind of linguistic annotation, whereas for the latter, some

[1] e.g. dependency relations that qualify *apple* might be 'object of *eat*' and 'adjective *red*'. This gives us dependency triples like $< apple, obj, eat >$.

form of syntactic annotation is needed. Also note that syntax-based approaches typically do not use any form of dimensionality reduction; using these seems much more cumbersome with syntax-based approaches, and does not seem to yield very sensible semantic dimensions.

Especially the syntax-based method has been adopted by many researchers in order to find semantically similar words. There is, however, one important problem with this kind of approach: the method is not able to cope with ambiguous words. Take the examples:

(1) een oneven nummer
a odd number
an odd number

(2) een steengoed nummer
a great number
'a great song'

The word *nummer* does not have the same meaning in these examples. In example (1), *nummer* is used in the sense of 'designator of quantity'. In example (2), it is used in the sense of 'musical performance'. Accordingly, we would like the word *nummer* to be disambiguated into two senses, the first sense being similar to words like *getal* 'number', *cijfer* 'digit' and the second to words like *liedje* 'song', *song* 'song'.

While it is relatively easy for a human language user to distinguish between the two senses, this is a difficult task for a computer. Even worse: the results get blurred because the attributes of both senses (in this example *oneven* and *steengoed*) are grouped together into one sense. This is the main drawback of the syntax-based method. On the other hand, methods that capture semantic dimensions are known to be useful in disambiguating different senses of a word. Particularly, probabilistic latent semantic analysis (plsa) is known to simultaneously encode various senses of words according to latent semantic dimensions [3]. In this paper, we want to explore an approach that tries to remedy the shortcomings of the former, syntax-based approach with the benefits of the latter. The intuition in this is that the syntactic features of the syntax-based approach can be disambiguated by the 'latent semantic dimensions' found in the window-based approach.

2 Previous Work

2.1 Distributional Similarity

There have been numerous approaches for computing the similarity between words from distributional data. We mention some of the most important ones.

One of the best known techniques is latent semantic analysis (lsa) [4, 5]. In lsa, a term-document matrix is created, containing the frequency of each word in a specific document. This matrix is then decomposed into three other matrices with a mathematical technique called singular value decomposition. The most important dimensions that come out of the SVD allegedly represent 'latent semantic dimensions', according to which nouns and documents can be presented more efficiently. Originally, LSA uses a term-document matrix, but subsequent researchers (e.g. [6]) have applied the same methods using bag of words co-occurrence information. In this view, LSA is an example of the window-based approach.[2]

LSA has been criticized for not being the most appropriate data reduction method for textual applications. The SVD underlying the method assumes normally-distributed data, whereas textual count data (such as the term-document matrix) can be more appropriately modeled by other distributional models such as Poisson [1, §15.4.3]. Successive methods such as PROBABILISTIC LATENT SEMANTIC ANALYSIS (PLSA) [3], try to remedy this shortcoming by imposing a proper latent variable model, according to which the values can be estimated. The method we adopt in our research – NON-NEGATIVE MATRIX FACTORIZATION – is similar to PLSA, and adequately remedies this problem as well.

The second approach – using syntactic relations – has been adopted by many researchers, in order to acquire semantically similar words. One of the most important is Lin's [8]. For Dutch, the approach has been applied by Van der Plas & Bouma [9].

2.2 Discriminating senses

Schütze [6] uses a disambiguation algorithm – called context-group discrimination – based on the clustering of the context of ambiguous words. The clustering is based on second-order co-occurrence: the contexts of the ambiguous word are similar if the words they in turn co-occur with are similar.

Pantel [10] presents a clustering algorithm – coined CLUSTERING BY COMMITTEE (CBC) – that automatically discovers word senses from text. The key idea is to first discover a set of tight, unambiguous clusters, to which possibly ambiguous words can be assigned. Once a word has been assigned to a cluster, the features associated with that particular cluster are stripped off the word's vector. This way, less frequent senses of the word can be discovered.

The former approach uses a window-based method; the latter uses syntactic data. But none of the algorithms developed so far have combined both sources in order to discriminate among different senses of a word.

3 Methodology

3.1 Non-negative Matrix Factorization

Theory Non-negative matrix factorization (NMF) [11] is a group of algorithms in which a matrix V is factorized into two other matrices, W and H.

[2] Note, however, that researchers have found substantial differences with regard to semantic similarity between document frequency and window-based co-occurrence frequency. See e.g. [7].

$$V_{n\times m} \approx W_{n\times r}H_{r\times m} \tag{1}$$

Typically r is much smaller than n, m so that both instances and features are expressed in terms of a few components.

Non-negative matrix factorization enforces the constraint that all three matrices must be non-negative, so all elements must be greater than or equal to zero. The factorization turns out to be particularly useful when one wants to find 'additive properties'.

Formally, the non-negative matrix factorization is carried out by minimizing an objective function. Two kinds of objective function exist: one that minimizes the Euclidian distance, and one that minimizes the Kullback-Leibler divergence. In this framework, we will adopt the latter, as – from our experience – entropy-based measures tend to work well for natural language. Thus, we want to find the matrices W and H for which the Kullback-Leibler divergence between V and WH (the multiplication of W and H) is the smallest.

Practically, the factorization is carried out through the iterative application of update rules. Matrices W and H are randomly initialized, and the rules in 2 and 3 are iteratively applied – alternating between them. In each iteration, each vector is adequately normalized, so that all dimension values sum to 1. The rules in 2 and 3 are guaranteed to converge to a local optimum in the minimization of the KL-divergence (for a proof, see [11]).

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} \frac{V_{i\mu}}{(WH)_{i\mu}}}{\sum_k W_{ka}} \tag{2}$$

$$W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} \frac{V_{i\mu}}{(WH)_{i\mu}}}{\sum_v H_{av}} \tag{3}$$

Example We can now straightforwardly apply NMF to create semantic word models. NMF is applied to a frequency matrix, containing bag of words co-occurrence data. The additive property of NMF ensures that semantic dimensions emerge, according to which the various words can be classified. Two sample dimensions are shown in example (3). For each dimension, the words with the largest value on that dimension are given. Dimension (a) can be qualified as a 'transport' dimension, and dimension (b) as a 'cooking' dimension.

(3) a. *bus* 'bus', *taxi* 'taxi', *trein* 'train', *halte* 'stop', *reiziger* 'traveler', *perron* 'platform', *tram* 'tram', *station* 'station', *chauffeur* 'driver', *passagier* 'passenger'

b. *bouillon* 'broth', *slagroom* 'cream', *ui* 'onion', *eierdooier* 'egg yolk', *laurierblad* 'bay leaf', *zout* 'salt', *deciliter* 'decilitre', *boter* 'butter', *bleekselderij* 'celery', *saus* 'sauce'

3.2 Extending Non-negative Matrix Factorization

We now propose an extension of NMF that combines both the bag of words approach and the syntactic approach. The algorithm finds again latent semantic dimensions, according to which nouns, contexts and syntactic relations are classified.

Since we are interested in the classification of nouns according to both 'bag-of-words' context and syntactic context, we first construct three matrices that capture the co-occurrence frequency information for each mode. The first matrix contains co-occurrence frequencies of nouns cross-classified by dependency relations, the second matrix contains co-occurrence frequencies of nouns cross-classified by words that appear in the noun's context window, and the third matrix contains co-occurrence frequencies of dependency relations cross-classified by co-occurring context words.

We then apply NMF to the three matrices, but we interleave the separate factorizations: the results of the former factorization are used to initialize the factorization of the next matrix. This implies that we need to initialize only three matrices at random; the other three are initialized by calculations of the previous step. The process is represented graphically in figure 1.

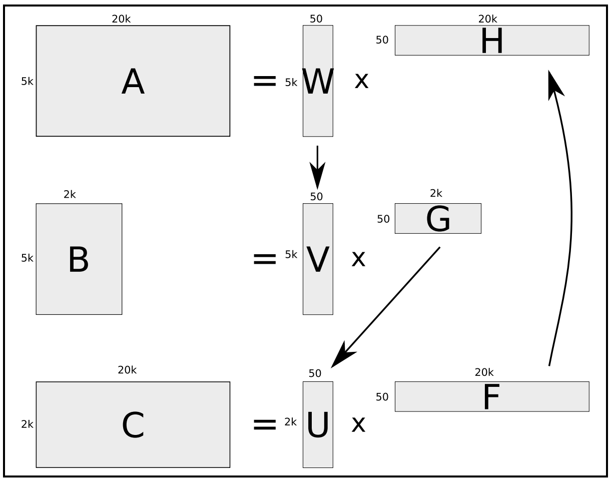

Fig. 1. A graphical representation of the extended NMF

When the factorization is finished, the three modes (nouns, dependency relations and context words) are classified according to latent semantic dimensions.

3.3 Sense Subtraction

Next, we want to use the factorization that has been created in the former step for word sense discrimination. The intuition is that we 'switch off' one dimension of an ambiguous word, to reveal possible other senses of the word. This intuition is implemented as follows:

From matrix W, we know which dimensions are the most important ones for a certain word. Matrix H gives us the importance of each syntactic relation given a dimension. By applying the formula in equation (4), we can 'subtract' the syntactic relations that are responsible for a certain dimension, from the vector in our original matrix.

$$\overrightarrow{v}_{new} = \overrightarrow{v}_{orig}(\overrightarrow{1} - \overrightarrow{h}_{dim}) \qquad (4)$$

This formula multiplies each feature (syntactic relation) of the original noun vector ($\overrightarrow{v}_{orig}$) with a scaling factor, according to the load of the feature on the subtracted dimension ($\overrightarrow{h}_{dim}$ – the vector of matrix H containing the dimension we want to subtract). $\overrightarrow{1}$ designates a vector of 1's the size of $\overrightarrow{h}_{dim}$.

4 Results

This section will show some exploratory results of three way contexts, and word sense discrimination using those contexts. It tends to give an idea of the kind of data produced, and the way word senses can be discriminated using these data. No formal evaluation of the method has been carried out yet.

4.1 Experimental Design

The interleaved NMF presented in section 3.2 has been applied to Dutch, using the CLEF corpus (containing Dutch newspaper text from 1994 and 1995). The corpus is consistently divided into paragraphs, which have been used as the context window for the bag-of-words mode. The corpus has been parsed by the Dutch dependency parser Alpino [12], and dependency triples have been extracted using XML-stylesheets. Next, the three matrices needed for our method have been constructed: one containing nouns by dependency relations (5K × 20K), one containing nouns by context words (5K × 2K) and one containing dependency relations by context words (20K × 2K). We did 50 iterations of the algorithm, factorizing the matrices into 50 dimensions. The NMF algorithm has been implemented in Python, using the NUMPY module for scientific computing.

4.2 Examples

In (4), an example is given of the kind of semantic dimensions found. This dimension may be called the 'transport' dimension, as is shown by the top 10 nouns (a), context words (b) and syntactic relations (c).

(4) a. *auto* 'car', *wagen* 'car', *tram* 'tram', *motor* 'motorbike', *bus* 'bus', *metro* 'subway', *automobilist* 'driver', *trein* 'trein', *stuur* 'steering wheel', *chauffeur* 'driver'

b. *auto* 'car', *trein* 'train', *motor* 'motorbike', *bus* 'bus', *rij* 'drive', *chauffeur* 'driver', *fiets* 'bike', *reiziger* 'reiziger', *passagier* 'passenger', *vervoer* 'transport'

c. viertraps$_{adj}$ 'four pedal', verplaats_met$_{obj}$ 'move with', toeter$_{adj}$ 'honk', tank_in_houd$_{obj}$ [parsing error], tank$_{subj}$ 'refuel', tank$_{obj}$ 'refuel', rij_voorbij$_{subj}$ 'pass by', rij_voorbij$_{adj}$ 'pass by', rij_af$_{subj}$ 'drive off', peperduur$_{adj}$ 'very expensive'

In what follows, we will talk about dimensions like this one as, e.g., the 'music' dimension or the 'city' dimension. In the vast majority of the cases, the dimensions are indeed as clear-cut as the transport dimension shown above, so that the dimensions can be rightfully labeled this way.

Next, two examples are given of how the semantic dimensions that have been found might be used for word sense discrimination. We will consider two ambiguous nouns: *pop*, which can mean 'pop music' as well as 'doll', and *Barcelona*, designating either the Spanish city or the Spanish football club.

First, we look up the top dimensions for each noun. Next, we subtract successively the highest and second highest dimension from the noun vector, as described in 3.3. This gives us three vectors for each noun: the original vector, and two vectors with one of the highest scoring dimensions eliminated. For each of these vectors, the top ten similar nouns are given, in order to compare the changes brought about.

(5) a. *pop*, *rock*, *jazz*, *meubilair* 'furniture', *popmuziek* 'pop music', *heks* 'witch', *speelgoed* 'toy', *kast* 'cupboard', *servies* '[tea] service', *vraagteken* 'question mark'

b. *pop*, *meubilair* 'furniture', *speelgoed* 'toy', *kast* 'cupboard', *servies* '[tea] service', *heks* 'witch', *vraagteken* 'question mark' *sieraad* 'jewel', *sculptuur* 'sculpture', *schoen* 'shoe'

c. *pop*, *rock*, *jazz*, *popmuziek* 'pop music', *heks* 'witch', *danseres* 'dancer', *servies* '[tea] service', *kopje* 'cup', *house* 'house music', *aap* 'monkey'

Example (5) shows the top similar words for the three vectors of *pop*. In (a), the most similar words to the original vector are shown. In (b), the top dimension (the 'music dimension') has been subtracted from (a), and in (c), the second highest dimension (a 'domestic items' dimension) has been subtracted from (a).

The differences between the three vectors are clear: in vector (a), both senses are mixed together, with 'pop music' and 'doll' items interleaved. In (b), no more music items are present. Only items related to the doll sense are among the top similar words. In (c), the music sense emerges much more clearly, with *rock*, *jazz* and *popmuziek* being the most similar, and a new music term (*house*) showing up among the top ten.

Admittedly, in vector (c), not all items related to the 'doll' sense are filtered out. We believe this is due to the fact that this sense cannot be adequately filtered out by one dimension (in this case, a dimension of 'domestic items' alone), whereas it is much easier to filter out the 'music' sense with only one 'music' dimension. In future work, we want to investigate the possibility of subtracting multiple dimensions related to one sense.

A second example, the ambiguous proper noun *Barcelona*, is given in (6).

(6) a. *Barcelona*, *Arsenal*, *Inter*, *Juventus*, *Vitesse*, *Milaan* 'Milan', *Madrid*, *Parijs* 'Paris', *Wenen* 'Vienna', *München* 'Munich'
b. *Barcelona*, *Milaan* 'Milan', *München* 'Munich', *Wenen* 'Vienna', *Madrid*, *Parijs* 'Paris', *Bonn*, *Praag* 'Prague', *Berlijn* 'Berlin', *Londen* 'London'
c. *Barcelona*, *Arsenal*, *Inter*, *Juventus*, *Vitesse*, *Parma*, *Anderlecht*, *PSV*, *Feyenoord*, *Ajax*

In (a), the two senses of *Barcelona* are clearly mixed up, showing cities as well as football clubs among the most similar nouns. In (b), where the 'football dimension' has been subtracted, only cities show up. In (c), where the 'city dimension' has been subtracted, only football clubs remain.

5 Conclusion & Future Work

In this paper, an extension of NMF has been presented that combines both bag of words data and syntactic data in order to find latent semantic dimensions according to which both words and syntactic relations can be classified. The use of three way data allows one to determine which dimension(s) are responsible for a certain sense of a word, and adapt the corresponding feature vector accordingly, 'subtracting' one sense to discover another one. We believe that the use of three way distributional data is effectively able to disambiguate the features of a given word, and accordingly its word senses.

We conclude with some issues for future work. First of all, we'd like to test the method that has been explored in this paper in a proper evaluation framework, and compare the method to other methods that discriminate senses. Next, we'd like to work out a proper probabilistic framework for the 'subtraction' of dimensions. And finally, we'd like to combine the method with a clustering approach. Thus, one can determine which are the important dimensions for a given cluster, subtract these from the individual words, and see whether other senses of the word emerge.

References

1. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, Massachussets (2000)
2. Harris, Z.: Distributional structure. In Katz, J.J., ed.: The Philosophy of Linguistics. Oxford University Press (1985) 26–47
3. Hofmann, T.: Probabilistic latent semantic analysis. In: Proc. of Uncertainty in Artificial Intelligence, UAI'99, Stockholm (1999)
4. Landauer, T., Dumais, S.: A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychology Review **104** (1997) 211–240

5. Landauer, T., Foltz, P., Laham, D.: An Introduction to Latent Semantic Analysis. Discourse Processes **25** (1998) 295–284
6. Schütze, H.: Automatic word sense discrimination. Computational Linguistics **24**(1) (1998) 97–123
7. Sahlgren, M.: The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. PhD thesis, Stockholm University (2006)
8. Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of COLING/ACL 98, Montreal, Canada (1998)
9. van der Plas, L., Bouma, G.: Syntactic contexts for finding semantically similar words. In van der Wouden, T., et al., eds.: Computational Linguistics in the Netherlands 2004. Selected Papers from the Fifteenth CLIN Meeting, Utrecht, LOT (2005) 173–184
10. Pantel, P., Lin, D.: Discovering word senses from text. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Special Interest Group on Knowledge Discovery in Data, ACM Press (2002) 613–619
11. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS. (2000) 556–562
12. van Noord, G.: At Last Parsing Is Now Operational. In Mertens, P., Fairon, C., Dister, A., Watrin, P., eds.: TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles, Leuven (2006) 20–42