MELODI: Semantic Similarity of Words and Compositional Phrases using Latent Vector Weighting

Tim Van de Cruys
IRIT, CNRS
tim.vandecruys@irit.fr

Stergos Afantenos
IRIT, Toulouse University
stergos.afantenos@irit.fr

Philippe Muller
IRIT, Toulouse University
philippe.muller@irit.fr

Abstract

In this paper we present our system for the SemEval 2013 Task 5a on semantic similarity of words and compositional phrases. Our system uses a dependency-based vector space model, in combination with a technique called latent vector weighting. The system computes the similarity between a particular noun instance and the head noun of a particular noun phrase, which was weighted according to the semantics of the modifier. The system is entirely unsupervised; one single parameter, the similarity threshold, was tuned using the training data.

1 Introduction

In the course of the last two decades, vector space models have gained considerable momentum for semantic processing. Initially, these models only dealt with individual words, ignoring the context in which these words appear. More recently, two different but related approaches emerged that take into account the interaction between different words within a particular context. The first approach aims at building a joint, compositional representation for larger units beyond the individual word level (e.g., the composed, semantic representation of the noun phrase *crispy chips*). The second approach, different but related to the first one, computes the specific meaning of a word within a particular context (e.g. the meaning of the noun *bank* in the context of the adjective *bankrupt*).

In this paper, we describe our system for the SemEval 2013 Task 5a: semantic similarity of words and compositional phrases – which follows the latter approach. Our system uses a dependency-based vector space model, in combination with a technique called latent vector weighting (Van de Cruys et al., 2011). The system computes the similarity between a particular noun instance and the head noun of a particular noun phrase, which was weighted according to the semantics of the modifier. The system is entirely unsupervised; one single parameter, the similarity threshold, was tuned using the training data.

2 Related work

In recent years, a number of methods have been developed that try to capture the compositional meaning of units beyond the individual word level within a distributional framework. One of the first approaches to tackle compositional phenomena in a systematic way is Mitchell and Lapata's (2008) approach. They explore a number of different models for vector composition, of which vector addition (the sum of each feature) and vector multiplication (the elementwise multiplication of each feature) are the most important. Baroni and Zamparelli (2010) present a method for the composition of adjectives and nouns. In their model, an adjective is a linear function of one vector (the noun vector) to another vector (the vector for the adjective-noun pair). The linear transformation for a particular adjective is represented by a matrix, and is learned automatically from a corpus, using partial least-squares regression. Coecke et al. (2010) present an abstract theoretical framework in which a sentence vector is a function of the Kronecker product of its word vectors, which allows for greater interaction between the different

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 98–102, Atlanta, Georgia, June 14-15, 2013.

word features. And Socher et al. (2012) present a model for compositionality based on recursive neural networks.

Closely related to the work on compositionality is research on the computation of word meaning in context. Erk and Padó (2008, 2009) make use of selectional preferences to express the meaning of a word in context. And Dinu and Lapata (2010) propose a probabilistic framework that models the meaning of words as a probability distribution over latent factors. This allows them to model contextualized meaning as a change in the original sense distribution.

Our work takes the latter approach of computing word meaning in context, and is described in detail below.

3 Methodology

Our method uses latent vector weighting (Van de Cruys et al., 2011) in order to compute a semantic representation for the meaning of a word within a particular context. The method relies upon a factorization model in which words, together with their window-based context features and their dependency-based context features, are linked to latent dimensions. The factorization model allows us to determine which dimensions are important for a particular context, and adapt the dependency-based feature vector of the word accordingly. The modified feature vector is then compared to the target noun feature vector with the cosine similarity function.

This following sections describe our model in more detail. In section 3.1, we describe non-negative matrix factorization – the factorization technique that our model uses. Section 3.2 describes our way of combining dependency-based context features and window-based context features within the same factorization model. Section 3.3, then, describes our method of computing the meaning of a word within a particular context.

3.1 Non-negative Matrix Factorization

Our latent model uses a factorization technique called non-negative matrix factorization (Lee and Seung, 2000) in order to find latent dimensions. The key idea is that a non-negative matrix $\mathbf{A}$ is factorized into two other non-negative matrices, $\mathbf{W}$ and $\mathbf{H}$

$$\mathbf{A}_{i\times j} \approx \mathbf{W}_{i\times k}\mathbf{H}_{k\times j} \tag{1}$$

where k is much smaller than i, j 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.

Using the minimization of the Kullback-Leibler divergence as an objective function, we want to find the matrices $\mathbf{W}$ and $\mathbf{H}$ for which the divergence between $\mathbf{A}$ and $\mathbf{WH}$ (the multiplication of $\mathbf{W}$ and $\mathbf{H}$) is the smallest. This factorization is carried out through the iterative application of update rules. Matrices $\mathbf{W}$ and $\mathbf{H}$ 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.

$$\mathbf{H}_{a\mu} \leftarrow \mathbf{H}_{a\mu}\frac{\sum_i \mathbf{W}_{ia}\frac{\mathbf{A}_{i\mu}}{(\mathbf{WH})_{i\mu}}}{\sum_k \mathbf{W}_{ka}} \tag{2}$$

$$\mathbf{W}_{ia} \leftarrow \mathbf{W}_{ia}\frac{\sum_\mu \mathbf{H}_{a\mu}\frac{\mathbf{A}_{i\mu}}{(\mathbf{WH})_{i\mu}}}{\sum_v \mathbf{H}_{av}} \tag{3}$$

3.2 Combining syntax and context words

Using an extension of non-negative matrix factorization (Van de Cruys, 2008), it is possible to jointly induce latent factors for three different modes: nouns, their window-based context words, and their dependency-based context features. The intuition is that the window-based context words inform us about broad, topical similarity, whereas the dependency-based features get at a tighter, synonym-like similarity. As input to the algorithm, two matrices are constructed that capture the pairwise co-occurrence frequencies for the different modes. The first matrix contains co-occurrence frequencies of words cross-classified by dependency-based features, and the second matrix contains co-occurrence frequencies of words cross-classified by words that appear in the word's context window. NMF is then applied to the two matrices, and the separate factorizations are interleaved (i.e. matrix $\mathbf{W}$, which contains the nouns by latent dimensions,

is shared between both factorizations). A graphical representation of the interleaved factorization algorithm is given in figure 1. The numbered arrows indicate the sequence of the updates.

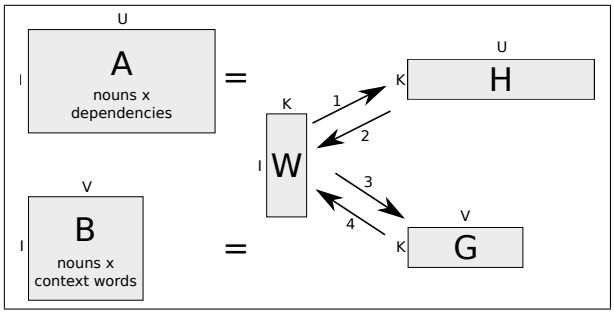

Figure 1: A graphical representation of the interleaved NMF

When the factorization is finished, the three different modes (words, window-based context words and dependency-based context features) are all represented according to a limited number of latent factors.

The factorization that comes out of the NMF model can be interpreted probabilistically (Gaussier and Goutte, 2005; Ding et al., 2008). More specifically, we can transform the factorization into a standard latent variable model of the form

$$p(w_i, d_j) = \sum_{z=1}^{K} p(z)p(w_i|z)p(d_j|z) \tag{4}$$

by introducing two $K \times K$ diagonal scaling matrices $\mathbf{X}$ and $\mathbf{Y}$, such that $\mathbf{X}_{kk} = \sum_i \mathbf{W}_{ik}$ and $\mathbf{Y}_{kk} = \sum_j \mathbf{H}_{kj}$. The factorization $\mathbf{WH}$ can then be rewritten as

$$\begin{aligned} \mathbf{WH} &= (\mathbf{WX}^{-1}\mathbf{X})(\mathbf{YY}^{-1}\mathbf{H}) \\ &= (\mathbf{WX}^{-1})(\mathbf{XY})(\mathbf{Y}^{-1}\mathbf{H}) \end{aligned} \tag{5}$$

such that $\mathbf{WX}^{-1}$ represents $p(w_i|\mathbf{z})$, $(\mathbf{Y}^{-1}\mathbf{H})^T$ represents $p(d_j|\mathbf{z})$, and $\mathbf{XY}$ represents $p(\mathbf{z})$. Using Bayes' theorem, it is now straightforward to determine $p(\mathbf{z}|d_j)$.

$$p(\mathbf{z}|d_j) = \frac{p(d_j|\mathbf{z})p(\mathbf{z})}{p(d_j)} \tag{6}$$

3.3 Meaning in Context

3.3.1 Overview

Using the results of the factorization model described above, we can now adapt a word's feature vector according to the context in which it appears. Intuitively, the context of the word (in our case, the dependency-based context feature that acts as an adjectival modifier to the head noun) pinpoint the important semantic dimensions of the particular instance, creating a probability distribution over latent factors. The required probability vector, $p(\mathbf{z}|d_j)$, is yielded by our factorization model. This probability distribution over latent factors can be interpreted as a semantic fingerprint of the passage in which the target word appears. Using this fingerprint, we can now determine a new probability distribution over dependency features given the context.

$$p(\mathbf{d}|d_j) = p(\mathbf{z}|d_j)p(\mathbf{d}|\mathbf{z}) \tag{7}$$

The last step is to weight the original probability vector of the word according to the probability vector of the dependency features given the word's context, by taking the pointwise multiplication of probability vectors $p(\mathbf{d}|w_i)$ and $p(\mathbf{d}|d_j)$.

$$p(\mathbf{d}|w_i, d_j) = p(\mathbf{d}|w_i) \cdot p(\mathbf{d}|d_j) \tag{8}$$

Note that this final step is a crucial one in our approach. We do not just build a model based on latent factors, but we use the latent factors to determine which of the features in the original word vector are the salient ones given a particular context. This allows us to compute an accurate adaptation of the original word vector in context.

3.3.2 Example

Let us exemplify the procedure with an example. Say we want to compute the distributionally similar words to the noun *instrument* within the phrases (1) and (2), taken from the task's test set:

(1) musical instrument

(2) optical instrument

First, we extract the context feature for both instances, in this case $C_1 = \{musical_{adj}\}$ for phrase (1), and $C_2 = \{optical_{adj}\}$ for phrase (2). Next, we

look up $p(\mathbf{z}|C_1)$ and $p(\mathbf{z}|C_2)$ – the probability distributions over latent factors given the context – which are yielded by our factorization model. Using these probability distributions over latent factors, we can now determine the probability of each dependency feature given the different contexts – $p(\mathbf{d}|C_1)$ and $p(\mathbf{d}|C_2)$ (equation 7).

The former step yields a general probability distribution over dependency features that tells us how likely a particular dependency feature is given the context that our target word appears in. Our last step is now to weight the original probability vector of the target word (the aggregate of dependency-based context features over all contexts of the target word) according to the new distribution given the context in which the target word appears (equation 8).

We can now return to our original matrix **A** and compute the top similar words for the two adapted vectors of *instrument* given the different contexts, which yields the results presented below.

1. **instrument**$_N$, C_1: *percussion*, *flute*, *violin*, *melody*, *harp*

2. **instrument**$_N$, C_2: *sensor*, *detector*, *amplifier*, *device*, *microscope*

3.4 Implementational details

Our model has been trained on the UKWaC corpus (Baroni et al., 2009). The corpus has been part of speech tagged and lemmatized with Stanford Part-Of-Speech Tagger (Toutanova and Manning, 2000; Toutanova et al., 2003), and parsed with MaltParser (Nivre et al., 2006) trained on sections 2-21 of the Wall Street Journal section of the Penn Treebank extended with about 4000 questions from the QuestionBank[1], so that dependency triples could be extracted.

The matrices needed for our interleaved NMF factorization are extracted from the corpus. Our model was built using 5K nouns, 80K dependency relations, and 2K context words[2] (excluding stop words) with highest frequency in the training set, which yields matrices of 5K nouns $\times$ 80K dependency relations, and 5K nouns $\times$ 2K context words.

[1] `http://maltparser.org/mco/english_parser/engmalt.html`

[2] We used a fairly large, paragraph-like window of four sentences.

model	accuracy	precision	recall	F1
dist	.69	.83	.48	.61
lvw	**.75**	**.84**	**.61**	**.71**

Table 1: Results of the distributional model (dist) and latent vector weighting model (lvw) on the SemEval task 5a

The interleaved NMF model was carried out using $K = 600$ (the number of factorized dimensions in the model), and applying 100 iterations. The interleaved NMF algorithm was implemented in Matlab; the preprocessing scripts and scripts for vector computation in context were written in Python.

The model is entirely unsupervised. The only parameter to set, the cosine similarity threshold ϕ, is induced from the training set. We set $\phi = .049$.

4 Results

Table 1 shows the evaluation results of the simple distributional model (which only takes into account the head noun) and our model that uses latent vector weighting. The results indicate that our model based on latent vector weighting performs quite a bit better than a standard dependency-based distributional model. The *lvw* model attains an accuracy of .75 – a 6% improvement over the distributional model – and an F-measure of .71 – a 10% improvement over the distributional model.

5 Conclusion

In this paper we presented an entirely unsupervised system for the assessment of the similarity of words and compositional phrases. Our system uses a dependency-based vector space model, in combination with latent vector weighting. The system computes the similarity between a particular noun instance and the head noun of a particular noun phrase, which was weighted according to the semantics of the modifier. Using our system yields a substantial improvement over a simple dependency-based distributional model, which only takes the head noun into account.

References

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA, October. Association for Computational Linguistics.

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

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

Chris Ding, Tao Li, and Wei Peng. 2008. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927.

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

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

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

Eric Gaussier and Cyril Goutte. 2005. Relation between PLSA and NMF and implications. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602, Salvador, Brazil.

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

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

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

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea, July. Association for Computational Linguistics.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pages 63–70.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, pages 252–259.

Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2011. Latent vector weighting for word meaning in context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1012–1022, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Tim Van de Cruys. 2008. Using three way data for word sense discrimination. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 929–936, Manchester.