Towards a Matrix-based Distributional Model of Meaning

Eugenie Giesbrecht
FZI Forschungszentrum Informatik
at the University of Karlsruhe
Haid-und-Neu-Str. 10-14, Karlsruhe, Germany
giesbrecht@fzi.de

Abstract

Vector-based distributional models of semantics have proven useful and adequate in a variety of natural language processing tasks. However, most of them lack at least one key requirement in order to serve as an adequate representation of natural language, namely sensitivity to structural information such as word order. We propose a novel approach that offers a potential of integrating order-dependent word contexts in a completely unsupervised manner by assigning to words characteristic distributional matrices. The proposed model is applied to the task of free associations. In the end, the first results as well as directions for future work are discussed.

1 Introduction

In natural language processing as well as in information retrieval, Vector Space Model (VSM) (Salton et al., 1975) and Word Space Model (WSM) (Schütze, 1993; Lund and Burgess, 1996) have become the mainstream for text representation. VSMs embody the distributional hypothesis of meaning, the main assumption of which is that a word is known "by the company it keeps" (Firth, 1957). VSMs proved to perform well in a number of cognitive tasks such as synonymy identification (Landauer and Dumais, 1997), automatic thesaurus construction (Grefenstette, 1994) and many others. However, it has been long recognized that these models are too weak to represent natural language to a satisfactory extent. With VSMs, the assumption is made that word co-occurrence is essentially independent of word order. All the co-occurrence information is thus fed into one vector per word.

Suppose our "background knowledge" corpus consists of one sentence: *Peter kicked the ball.* It follows that the distributional meanings of both PETER and BALL would be in a similar way defined by the co-occurring KICK which is insufficient, as BALL can be only *kicked* by somebody but not *kick* itself; in case of PETER, both ways of interpretation should be possible. To overcome the aforementioned problems with vector-based models, we suggest a novel distributional paradigm for representing text in that we introduce a further dimension into a "standard" two-dimensional word space model. That allows us to count correlations for three words at a time. In short, given a vocabulary V, context width $w = m$ and tokens $t_1, t_2, t_3, ..., t_i \in V$, for token t_i a matrix of size $V \times V$ is generated that has nonzero values in cells where t_i appears between t_{i-m} and t_{i+m}.

Note that this 3-dimensional representation allows us to integrate word order information into the model in a completely unsupervised manner as well as to achieve a richer word representation as a matrix instead of a vector.

The remainder of the paper is organized as follows. After a recap of basic mathematical notions and operations used in the model in Section 2, we introduce the proposed three-dimensional tensor-based model of text representation in Section 3. First evaluation experiments are reported in Section 4.

Proceedings of the NAACL HLT 2010 Student Research Workshop, pages 23–28,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

After a brief overview of related work in Section 5, we provide some concluding remarks and suggestions for future work in Section 6.

2 Preliminaries

In this section, we provide a brief introduction to tensors and the basics of mathematical operations that are employed in the suggested model.

First, given d natural numbers $n_1, \ldots, n_d$, a *(real)* $n_1 \times \ldots \times n_d$ *tensor* can be defined as a function $T : \{1, \ldots, n_1\} \times \ldots \times \{1, \ldots, n_d\} \rightarrow \mathbb{R}$, mapping d-tuples of natural numbers to real numbers. Intuitively, a tensor can best be thought of as a d-dimensional table (or array) carrying real numbers as entries. Thereby $n_1, \ldots, n_d$ determine the extension of the array in the different directions. Obviously, matrices can be conceived as $n_1 \times n_2$-tensors and vectors as n_1-tensors.

In our setting, we will work with tensors where $d = 3$ and for the sake of better understandability we will introduce the necessary notions for this case only.

Our work employs *higher-order singular value decomposition* (HOSVD), which generalizes the method of singular value decomposition (SVD) from matrices to arbitrary tensors.

Given an $n_1 \times n_2 \times n_3$ tensor T, its *Tucker decomposition* (Tucker, 1966) for given natural numbers m_1, m_2, m_3 consists of an $m_1 \times m_2 \times m_3$ tensor G and three matrices A, B, and C of formats $n_1 \times m_1$, $n_2 \times m_2$, and $n_3 \times m_3$, respectively, such that

$$T(i, j, k) = \sum_{r=1}^{m_1} \sum_{s=1}^{m_2} \sum_{t=1}^{m_3} G(r, s, t) \cdot A(i, r) \cdot B(j, s) \cdot C(k, t).$$

The idea here is to represent the large-size tensor T by the smaller "core" tensor G. The matrices A, B, and C can be seen as linear transformations "compressing" input vectors from dimension n_i into dimension m_i. Note that a precise representation of T is not always possible. Rather one may attempt to approximate T as well as possible, i.e. find the tensor T' for which a Tucker decomposition exists and which has the least distance to T. Thereby, the notion of distance is captured by $\|T - T'\|$, where $T - T'$ is the tensor obtained by entry-wise subtraction and $\| \cdot \|$ is the *Frobenius norm* defined by

$$\|M\| = \sqrt{\sum_{r=1}^{n_1} \sum_{s=1}^{n_2} \sum_{t=1}^{n_3} (M(r, s, t))^2}.$$

In fact, the described way of approximating a tensor is called *dimensionality reduction* and is often used for reducing noise in multi-dimensional data.

3 Proposed Model

Our motivation is to integrate structure into the geometrical representation of text meaning while adhering to the ideas of distributional semantics. For this, we introduce a third dimension that allows us to separate the left and right contexts of the words. As we process text, we accumulate the left and right word co-occurrences to represent the meaning of the current word. Formally, given a corpus $\mathcal{K}$, a list L of tokens, and a context width w, we define its tensor representation $T_{\mathcal{K}}$ by letting $T_{\mathcal{K}}(i, j, k)$ be the number of occurrences of $L(j)\ s\ L(i)\ s'\ L(k)$ in sentences in $\mathcal{K}$ where s, s' are (possibly empty) sequences of at most $w - 1$ tokens. For example, suppose our corpus consists of three sentences: "Paul kicked the ball slowly. Peter kicked the ball slowly. Paul kicked Peter." We let $w = 1$, presuming prior stop words removal. We obtain a $5 \times 5 \times 5$ tensor. Table 1 displays two i-slices of the resulting tensor T showing left vs. right context dependencies.

KICK	PETER	PAUL	KICK	BALL	SLOWLY
PETER	0	0	0	1	0
PAUL	1	0	0	1	0
KICK	0	0	0	0	0
BALL	0	0	0	0	0
SLOWLY	0	0	0	0	0

BALL	PETER	PAUL	KICK	BALL	SLOWLY
PETER	0	0	0	0	0
PAUL	0	0	0	0	0
KICK	0	0	0	0	2
BALL	0	0	0	0	0
SLOWLY	0	0	0	0	0

Table 1: Slices of T for the terms KICK ($i = 3$) and BALL ($i = 4$).

Similarly to traditional vector-based distributional models, dimensionality reduction needs to be performed in three dimensions either, as the resulting tensor is very sparse (see the examples of KICK and

BALL). To this end, we employ Tucker decomposition for 3 dimensions as introduced in Section 2. For this, Matlab Tensor Toolbox[1] (Bader and Kolda, 2006) is used.

A detailed overview of computational complexity of Tucker decomposition algorithms in Tensor Toolbox is provided in Turney (2007). The drawback of those is that their complexity is cubic in the number of factorization dimensions and unfeasible for large datasets. However, new memory efficient tensor decomposition algorithms have been proposed in the meantime. Thus, *Memory Efficient Tucker (MET)* is available in Matlab Tensor Toolbox since Version 2.3. Rendle and Schmidt-Thieme (2010) present a new factorization method with linear complexity.

4 Evaluation Issues

4.1 Task

Vector-based distributional similarity methods have proven to be a valuable tool for a number of tasks on automatic discovery of *semantic relatedness* between words, like synonymy tests (Rapp, 2003) or detection of analogical similarity (Turney, 2006).

A somewhat related task is the task of finding out to what extent (statistical) similarity measures correlate with free word associations[2]. Furthermore, this task was suggested as a *shared task* for the evaluation of word space models at Lexical Semantics Workshop at ESSLLI 2008. *Free associations* are the words that come to the mind of a native speaker when he or she is presented with a so-called *stimulus word*. The percent of test subjects that produce certain *response* to a given *stimulus* determines the degree of a free association between a *stimulus* and a *response*.

Despite the widespread usage of vector-based models to retrieve semantically similar words, it is still rather unclear what type of linguistic phenomena they model (cf. Heylen et al. (2008), Wandmacher et al. (2008)). The same is true for *free associations*. There are a number of relations according to which a word may be associated with another word. For example, Aitchison (2003) distinguishes four types of associations: co-ordination, collocation, superordination and synonymy. This affords an opportunity to use the task of *free associations* as a "baseline" for distributional similarity.

For this task, workshop organizers have proposed three subtasks, one of which - *discrimination* - we adapt in this paper. Test sets have been provided by the workshop organizers. The former are based on the Edinburgh Associative Thesaurus[3] (EAT), a freely available database of English association norms.

Discrimination task includes a test set of overall 300 word pairs that were classified according to three classes of association strengths:

- FIRST strongly associated word pairs as indicated by more than 50% of test subjects as first responses;
- HAPAX word associations that were produced by a single test subject;
- RANDOM random combinations of words from EAT that were never produced as a *stimulus - response* pair.

4.2 Procedure

To collect the three-way co-occurrence information, we experiment with the UKWAC corpus (A. Ferraresi and Bernardini, 2008), as suggested by the workshop organizers, in order to get comparable results. As UKWAC is a huge Web-derived corpus consisting of about 2 billion tokens, it was impossible at the current stage to process the whole corpus. As the subsections of UKWAC contain randomly chosen documents, one can train the model on any of the subsections.

We limited out test set to the word pairs for which the constituent words occur more than 50 times in the test corpus. Thereby, we ended up with a test set consisting of 222 word pairs.

We proceed in the following way. For *each pair of words*:

1. Gather N sentences, i.e. contexts, for each of the two words[4], here $N = 50$;

[1] Version 2.3

[2] One of the reasons to choose this evaluation setting was that the dataset for *free word associations task* is freely available at http://wordspace.collocations.de/doku.php/data:esslli2008:start (in contrast to, e.g., the synonymy test set).

[3] http://www.eat.rl.ac.uk/

[4] This corpus "preprocessing" step was mainly due to lim-

2. Build a 3-dimensional tensor from the subcorpus obtained in (1), given a context width w=5, i.e. 5 words to the left and 5 words to the right of the target word), taking sentence boundaries into consideration;

3. Reduce 5 times the dimensionality of the tensor obtained in (2) by means of Tucker decomposition;

4. Extract two matrices of both constituents of the word pair and compare those by means of cosine similarity[5].

Here, we follow the tradition of vector-based models where *cosine* is usually used to measure *semantic relatedness*. One of the future direction in matrix-based meaning representation is to investigate further matrix comparison metrics.

4.3 Results and Discussion

Tables 2 and 3 show the resulting accuracies[6] for training and test sets. th denotes cosine threshold values that were used for grouping the results. Here, th is taken to be the function of the size s of the data set. Thus, given a training set of size $s = 60$ and 3 classes, we define an "equally distributed" threshold $th_1 = 60/3 = 20$ (s. Table 2) and a "linearly growing" threshold $th_2 = \frac{1}{4}, \frac{1}{3}, rest$ (s. Table 3).

It is not quite apparent, how the threshold for differentiating between the groups should be determined under given conditions. Usually, such measures are defined on the basis of training data (e.g. Wandmacher et al. (2008)). It was not applicable in our case as, due to the current implementation of the model as well as insufficient computational resources for the time being, we could not build one big model for all experiment iterations.

Also, the intuition we have gained with this kind of thresholds is that as soon as you change the underlying corpus or the model parameters, you may need to define new thresholds (cf. Tables 2 and 3). Thresholds in geometric models of meaning can not be just fixed, just as the measure of similarity cannot be easily quantified by humans.

It would be straightforward to compare the performance of the proposed model with its 2-dimensional analogue. Wandmacher et al. (2008) obtain in average better results with their LSA-based model for this task. Specifically, they observe very good results for RANDOM associations (78.2% accuracy) but the lowest results for the FIRST, i.e. strongest, associations (50%). In constrast, the outcome for RANDOM in our model is the worst. However, the bigger the threshold, the more accurate is getting the model for the FIRST associations. For example, with a threshold of $th = 0.2$ for the test set - 4 out of 5 highest ranked pairs were highly associated (FIRST) and the fifth pair was from the HAPAX group. For HAPAX word associations, no similar regularities could be observed.

The resulting accuracies may seem to be poor at this stage. However, it is worth mentioning that this is a highly difficult and corpus-dependent task for automatic processing. The reported results have been obtained based on very small corpora, containing ca. 100 sentences per iteration (cf. Wandmacher et al. (2008) use a corpus of 108 million words to train their LSA-Model). Consequently, it is not possible to compare both results directly, as they have been produced under very different conditions.

5 Related Work

5.1 Matrix Approaches

There have been a number of efforts to integrate syntax into vector-based models with alternating success. Some used (dependency) parsing to feed the models (Grefenstette, 1994; Lin, 1998; Padó and Lapata, 2007); the others utilized only part of speech information, e.g., Widdows (2003).

In many cases, these syntactically enhanced models improved the performance (Grefenstette, 1994; Lin, 1998; Padó and Lapata, 2007). Sometimes, however, rather controversial results were observed. Thus, Widdows (2003) reported both positive and negative effects for the task of developing taxonomies. On the one side, POS information increased the performance for common nouns; on the other side, it degraded the outcome for proper nouns

ited processing power we had at our disposal at the moment the experiments were conducted. With this step, we considerably reduced the size of the corpus and guaranteed a certain number of contexts per relevant word.

[5] Cosine similarity is determined as a normalized inner product

[6] Accuracy is defined in the following way: $Accuracy = right/(right + wrong)$

	TRAIN	TEST
FIRST	12/20 (60%) ($th = 0.022$)	25/74 (33%) ($th = 0.078$))
HAPAX	7/20 (35%) ($th = 0.008$)	35/74 (47%) $th = 0.042$)
RANDOM	8/20 (40%)	23/74 (31%)
TOTAL (F/H/R)	27/60 (45%)	83/222 (37.4%)
FIRST/HORR[7]	44/60 (73.33%)	125/222 (56.3%)

Table 2: Accuracies for the “equally distributed” threshold for training and test sets

	TRAIN	TEST
FIRST	9/15 (60%) ($th = 0.0309$)	20/55 (36.4%) ($th = 0.09$)
HAPAX	8/20 (40%) ($th = 0.0101$)	39/74 (52.7%) ($th = 0.047$)
RANDOM	10/25 (40%)	24/93 (25.8%)
TOTAL (F/H/R)	27/60 (45%)	108/222 (48.6%)
FIRST/HORR[8]	43/60 (71.60%)	113/222 (50.9%)

Table 3: Accuracies for a “linearly growing” threshold for training and test sets

and verbs.

Sahlgren et al. (2008) incorporate word order information into context vectors in an unsupervised manner by means of permutation.

Recently, Erk and Padó (2008) proposed a structured vector space model where a word is represented by several vectors reflecting the words lexical meaning as well as its selectional preferences. The motivation behind their work is very close to ours, namely, that single vectors are too weak to represent word meaning. However, we argue that a matrix-based representation allows us to integrate contextual information in a more general manner.

5.2 Tensor Approaches

Among the early attempts to apply higher-order tensors instead of vectors to text data is the work of Liu et al. (2005) who show that Tensor Space Model is consistently better than VSM for text classification. Cai et al. (2006) suggest a 3-dimensional representation for documents and evaluate the model on the task of document clustering.

The above as well as a couple of other projects in this area in information retrieval community leave open the question of h*ow to convey text into a three-dimensional tensor*. They still use vector-based representation as the basis and then just mathematically convert vectors into tensors, without linguistic justification of such transformations.

Further, there are few works that extend the term-document matrix with metadata as a third dimension (Chew et al., 2007; Sun et al., 2006).

Turney (2007) is one of the few to study the application of tensors to word space models. However, the emphasis in that paper is more on the evaluation of different tensor decomposition models for such spaces than on the formal model of text representation in three dimensions. Van de Cruys (2009) suggests a three-way model of co-occurrence similar to ours. In contrast to Van de Cruys (2009), we are not using any explicit syntactic preprocessing. Furthermore, our focus is more on the model itself as a general model of meaning.

6 Summary and Future Work

In this paper, we propose a novel approach to text representation inspired by the ideas of distributional semantics. In particular, our model suggests a solution to the problem of integrating word order information in vector spaces in an unsupervised manner. First experiments on the task of free associations are reported. However, we are not in the position yet to commit ourselves to any representative statements. A thorough evaluation of the model still needs to be done. Next steps include, amongst others, evaluating the suggested model with a bigger data corpus as well as using stemming and more sophisticated filling of word matrices, e.g., by introducing advanced weighting schemes into the matrices instead of simple counts.

Furthermore, we started with evaluation on the task which has been proposed for the evaluation of word space models at the level of word meaning. We need, however, to evaluate the model for the tasks where word order information matters more, e.g. on selectional preferences or paraphrasing.

Last but not least, we plan to address the issue of modeling compositional meaning with matrix-based distributional model of meaning.

Acknowledgments

This work is supported by German "Federal Ministry of Economics" (BMWi) under the project Theseus (number 01MQ07019). Many thanks to the anonymous reviewers for their insightful comments.

References

M. Baroni A. Ferraresi, E. Zanchetta and S. Bernardini. 2008. Introducing and evaluating ukWaC, a very large Web-derived corpus of English. In *Proceedings of the WAC4 Workshop at LREC'08*.

Jean Aitchison. 2003. *Words in the Mind: An Introduction to the Mental Lexicon*. Wiley-Blackwell.

Brett W. Bader and Tamara G. Kolda. 2006. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Trans. Math. Softw.*, 32(4):635–653.

Deng Cai, Xiaofei He, and Jiawei Han. 2006. Tensor space model for document analysis. In *SIGIR*, pages 625–626. ACM.

Peter Chew, Brett Bader, Tamara Kolda, and Ahmed Abdelali. 2007. Cross-language information retrieval using PARAFAC2. In *Proc. KDD'07*, pages 143–152. ACM.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *EMNLP*, pages 897–906. ACL.

J.R. Firth. 1957. A synopsis of linguistic theory 1930-55. *Studies in linguistic analysis*, pages 1–32.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Springer.

Kris Heylen, Yves Peirsman, Dirk Geeraerts, and Dirk Speelman. 2008. Modelling word similarity: an evaluation of automatic synonymy extraction algorithms. In *Proceedings of LREC'08*, pages 3243–3249.

T. K. Landauer and S. T Dumais. 1997. Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge. *Psychological Review*, 104(2):211–240.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of ACL'98*, pages 768–774. ACL.

Ning Liu, Benyu Zhang, Jun Yan, Zheng Chen, Wenyin Liu, Fengshan Bai, and Leefeng Chien. 2005. Text representation: from vector to tensor. In *Proc. ICDM05*.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instrumentation, and Computers*, pages 203–20.

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

Reinhard Rapp. 2003. Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, pages 315–322.

Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90, New York, NY, USA. ACM.

M. Sahlgren, A. Holst, and P. Kanerva. 2008. Permutations as a means to encode order in word space. In *Proc. CogSci08*, pages 1300–1305.

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Hinrich Schütze. 1993. Word space. In *Advances in NIPS 5*, pages 895–902.

J. Sun, D. Tao, and C. Faloutsos. 2006. Beyond streams and graphs: Dynamic tensor analysis. In *Proc. KDD'06*, pages 374–383.

L.R. Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3).

Peter D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

P. Turney. 2007. Empirical evaluation of four tensor decomposition algorithms. Technical report. Technical Report ERB-1152.

Tim Van de Cruys. 2009. A non-negative tensor factorization model for selectional preference induction. In *GEMS '09: Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 83–90, Morristown, NJ, USA. ACL.

Tonio Wandmacher, Ekaterina Ovchinnikova, and Theodore Alexandrov. 2008. Does Latent Semantic Analysis reflect human associations. In *Proceedings of the Lexical Semantics workshop at ESSLLI*, Hamburg, Germany.

Dominic Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of NAACL'03*, pages 197–204. ACL.